UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PÓS-EDIÇÃO AUTOMÁTICA DE TEXTOS TRADUZIDOS AUTOMATICAMENTE DE INGLÊS PARA PORTUGUÊS DO BRASIL

DÉBORA BEATRIZ DE JESUS MARTINS

ORIENTADOR: HELENA DE MEDEIROS CASELI

São Carlos – SP

Fevereiro de 2014

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PÓS-EDIÇÃO AUTOMÁTICA DE TEXTOS TRADUZIDOS AUTOMATICAMENTE DE INGLÊS PARA PORTUGUÊS DO BRASIL

DÉBORA BEATRIZ DE JESUS MARTINS

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

Orientador: Helena de Medeiros Caseli

São Carlos – SP

Fevereiro de 2014

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

M386pe

Martins, Débora Beatriz de Jesus.
Pós-edição automática de textos traduzidos automaticamente de inglês para português do Brasil / Débora Beatriz de Jesus Martins. -- São Carlos : UFSCar, 2014.
97 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2014.

1. Ciência da computação. 2. Linguagem - tradução automática. 3. Aprendizado de computador. 4. Identificação. I. Título.

CDD: 004 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

"Pós-Edição Automática de Textos Traduzidos Automaticamente de Inglês para Português do Brasil"

Débora Beatriz de Jesus Martins

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

Membros da Banca:

Helena de M. Caseli

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

Profa. Dra. Maria das Graças Volpe Nunes
(ICMC/USP)

Prof. Dr. Fábio Alves
(UFMG)

São Carlos
Abril/2014

À minha filha, por ter chegado ao mundo durante este mestrado, trazendo um amor à minha vida que eu nunca havia pensado ser possível sentir por alguém, fato que me deu ainda mais ânimo e motivação para continuar trabalhando e dedicar este projeto a ela.

AGRADECIMENTOS

À minha mãe que, sempre que eu precisava, me socorria e, mesmo não entendendo de computação, escutava descrições detalhadas dos meus experimentos com muita dedicação e carinho.

Ao meu pai, uma inspiração, por sua luta e história de vida, que desde muito jovem perseverou para conseguir o que tem e tornar-se o homem que é hoje.

Ao meu marido, meu grande amor, pela paciência e carinho que teve por mim nas horas mais difíceis.

À minha irmã, Daniele, que mesmo estando distante, sempre me encorajou a perseguir meus objetivos.

À Amanda, irmã querida.

Aos meus colegas no Lalic, Thiago Vieira e Leonardo Taba, que sempre me ajudaram quando eu precisei.

À Helena, minha orientadora, que hoje considero uma amiga, sempre muito atenciosa, dedicada ao projeto e compreensiva.

Aos professores Letícia Rezende e Bento Dias da Silva, da Unesp e Thiago Pardo, da USP, que me deram aulas como aluna especial e, através de sua paixão por ensinar, me motivaram a buscar este mestrado.

À UFSCar, ao LaLiC, à Unesp e ao CELiC pelas instalações.

À FAPESP (processo número 2011/03799-4) pelo apoio sem o qual não seria possível o desenvolvimento deste trabalho.

RESUMO

O projeto de mestrado descrito neste documento tem como foco a pós-edição de textos traduzidos automaticamente. Tradução Automática (TA) é a tarefa de traduzir textos em língua natural desempenhada por um computador e faz parte da linha de pesquisa de Processamento de Línguas Naturais (PLN), vinculada à área de Inteligência Artificial (IA). As pesquisas em TA, utilizando desde abordagens linguísticas até modelos estatísticos, têm avançado muito desde seu início na década de 1950. Entretanto, os textos traduzidos automaticamente, exceto quando utilizados apenas para um entendimento geral do assunto, ainda precisam passar por pós-edição para que se tornem bem escritos na língua alvo. Atualmente, a forma mais comum de pós-edição é a executada por tradutores humanos, sejam eles profissionais ou os próprios usuários dos sistemas de TA. A pós-edição manual é mais precisa, mas traz custo e demanda tempo, especialmente quando envolve muitas alterações. Como uma tentativa para avançar o estado-da-arte das pesquisas em TA, principalmente envolvendo o português do Brasil, esta pesquisa visa verificar a efetividade do uso de um sistema de pós-edição automática (*Automated Post-Editing* ou APE) na tradução do inglês para o português. Utilizando um corpus de treinamento contendo traduções de referência (boas traduções produzidas por humanos) e traduções geradas por um sistema de TA estatística baseada em frases, técnicas de aprendizado de máquina foram aplicadas para o desenvolvimento do APE. O sistema de APE desenvolvido: (i) identifica automaticamente os erros de TA e (ii) realiza a correção automática da tradução com ou sem a identificação prévia dos erros. A avaliação foi realizada usando tanto medidas automáticas BLEU e NIST, calculadas para as sentenças sem e com a pós-edição; como análise manual. Apesar de resultados limitados pelo pequeno tamanho do corpus de treinamento, foi possível concluir que o APE desenvolvido melhora a qualidade da TA de inglês para português.

Palavras-chave: Tradução Automática, Aprendizado de Máquina, Identificação Automática de Erros de Tradução, Pós-edição automática

ABSTRACT

The project described in this document focusses on the post-editing of automatically translated texts. Machine Translation (MT) is the task of translating texts in natural language performed by a computer and it is part of the Natural Language Processing (NLP) research field, linked to the Artificial Intelligence (AI) area. Researches in MT using different approaches, such as linguistics and statistics, have advanced greatly since its beginning in the 1950's. Nonetheless, the automatically translated texts, except when used to provide a basic understanding of a text, still need to go through post-editing to become well written in the target language. At present, the most common form of post-editing is that executed by human translators, whether they are professional translators or the users of the MT system themselves. Manual post-editing is more accurate but it is cost and time demanding and can be prohibitive when too many changes have to be made. As an attempt to advance in the state-of-the-art in MT research, mainly regarding Brazilian Portuguese, this research has as its goal verifying the effectiveness of using an Automated Post-Editing (APE) system in translations from English to Portuguese. By using a training corpus containing reference translations (good translations produced by humans) and translations produced by a phrase-based statistical MT system, machine learning techniques were applied for the APE creation. The resulting APE system is able to: (i) automatically identify MT errors and (ii) automatically correct MT errors by using previous error identification or not. The evaluation of the APE effectiveness was made through the usage of the automatic evaluation metrics BLEU and NIST, calculated for post-edited and not post-edited sentences. There was also manual verification of the sentences. Despite the limited results that were achieved due to the small size of our training corpus, we can conclude that the resulting APE improves MT quality from English to Portuguese.

Keywords: Machine Translation, Machine Learning, Automated Translation Error Identification, Automated Post-Editing

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Capítulo 1

INTRODUÇÃO

No mundo globalizado, a quantidade de informação disponível digitalmente, sobre os mais variados temas, aumenta mais de 100% a cada dois anos (LOHR, 2012). Grande parte dessa informação disponível em manuais, fóruns, reportagens, *blogs*, mensagens de correio eletrônico, comentários em redes sociais, etc. é representada em língua natural usando diferentes idiomas, o que coloca em destaque pesquisas na área de PLN (Processamento de Língua Natural), com especial atenção para a Tradução Automática.

Tradução pode ser entendida como o processo que envolve a compreensão de um texto em uma língua fonte e a elaboração de sua representação correspondente na língua alvo desejada. A tradução pode ser executada por humanos, automaticamente, ou uma combinação de ambos. A Tradução Automática (TA), ou *Machine Translation* (MT), é a tarefa de traduzir textos em língua natural desempenhada por um computador.

Segundo Silva (2010), as investigações sobre tradução automática (TA) tiveram início na década de 1950, onde a sistematização computacional das classes de palavras descritas nos manuais de gramática tradicional e a identificação computacional de poucos tipos de constituintes oracionais eram os objetos de estudo. Na atualidade, apesar do desenvolvimento de sistemas complexos integrando conhecimento linguístico ou não-linguístico, além de abordagens estatísticas avançadas e manipulação de imensos corpora[1], ainda há vários pontos a serem refinados no estudo da TA. Entre os idiomas que ainda carecem de mais pesquisas está o português do Brasil, idioma principal sob investigação neste trabalho.

Em mais de 60 anos de pesquisas em TA foram produzidos diversos sistemas e ferramentas comerciais ou disponíveis *on-line*, de código aberto ou não, como o Systran[2], o Apertium[3]

[1]Um corpus (cujo plural é corpora) é uma coletânea de textos escritos em uma língua. Essa coletânea serve como base de exemplos para aprendizado em diversas áreas de PLN (Processamento de Língua Natural). Quando a coletânea é formada por pares de textos em idiomas diferentes, sendo um texto a tradução do outro, ela recebe o nome de corpus paralelo.

[2]Disponível em: <http://www.systransoft.com/>. Acesso em: 17 jan. 2014.

[3]Disponível em: <http://www.apertium.org/>. Acesso em: 17 jan. 2014.

e os tradutores do Google[4], sendo tais sistemas desenvolvidos seguindo várias abordagens de TA. Porém, mesmo com esses esforços, ainda não foi possível alcançar as ambiciosas metas impostas no surgimento da TA: produzir TA de boa qualidade em domínios irrestritos, por meio de sistemas completamente automáticos. Assim, a tarefa de produzir textos de alta qualidade traduzidos automaticamente ainda é um grande desafio para a área de PLN.

Na atualidade, os textos traduzidos automaticamente, com algumas exceções quando utilizados apenas para um entendimento geral do assunto, precisam passar por um processo de pós-edição para que se tornem mais inteligíveis e bem escritos na língua alvo. De acordo com Krings (2001), o processo de pós-edição da TA é normalmente executado por um humano que faz modificações no texto traduzido com o intuito de transformá-lo em um texto aceitável para o propósito desejado. Para executar tal tarefa, o humano faz uma comparação do texto fonte (original) com a saída do tradutor automático. Essa pós-edição geralmente é executada por especialistas em tradução, o que gera um alto custo; ou pelos próprios usuários de um sistema de TA, o que demanda bastante tempo e nem sempre produz a saída desejada.

Nesse sentido, a pesquisa descrita neste documento surge como uma alternativa para a pós-edição completamente manual da TA propondo a aplicação de técnicas de aprendizado automático (Aprendizado de Máquina, AM) para automatizar o processo de pós-edição. Por meio da aplicação de técnicas de AM a um conjunto de treinamento composto por textos traduzidos automaticamente acompanhados de suas versões corretas, é possível aprender a identificar e corrigir os erros de tradução. Ao final do processo de treinamento, aplica-se o conhecimento de "como corrigir a TA" na pós-edição automática de textos traduzidos automaticamente. A automatização da pós-edição e o próprio sistema resultante recebem o nome de APE, do inglês *automated post-editing/editor.*

1.1 Motivação

Apesar de mais de 60 anos de esforços na produção de tradutores automáticos, a qualidade da TA como produto final ainda não atingiu os patamares desejados. Em (CASELI, 2007) foram verificados diversos trabalhos que analisavam sistemas de TA existentes para o português do Brasil (e os idiomas inglês e espanhol), idioma principal sob investigação nesta pesquisa. Em todos eles as sentenças incorretamente traduzidas ultrapassavam 50% do total.

Nesta pesquisa, os números encontrados na anotação do corpus de treinamento inglês-português (detalhada no capítulo 3) apontam que atualmente persiste o cenário e que ainda há muito a ser melhorado. Para a geração do corpus de treinamento foi utilizado um tradutor automático inglês-português que segue a abordagem considerada o estado-da-arte segundo medi-

[4]Disponível em: <http://translate.google.com.br/>. Acesso em: 17 jan. 2014.

das automáticas de avaliação como BLEU (PAPINENI et al., 2002) e NIST (DODDINGTON, 2002): a tradução automática estatística baseada em frases. Como resultado da anotação, constatou-se que 67% das sentenças traduzidas para o português do Brasil apresentaram um ou mais erros.

O gráfico da Figura 1.1 mostra as porcentagens de ocorrência das categorias de erro (detalhadas na seção 3.2) anotadas no corpus de treinamento. Pode-se notar que os "erros lexicais" (como palavras ausentes, extras, não traduzidas ou incorretamente traduzidas) são os mais comuns (44,48%), seguidos dos "erros sintáticos" (como erros de concordância em gênero, em número ou de flexão verbal) (38,61%). Detalhando em subcategorias, os erros mais anotados foram os de "palavra ausente" (14,81%) seguidos de "palavra incorretamente traduzida" (14,16%) e de erros de concordância, tanto em número (14,02%), como em gênero (12,62%). As porcentagens de ocorrência das subcategorias de erro anotadas podem ser vistas na Figura 1.2.

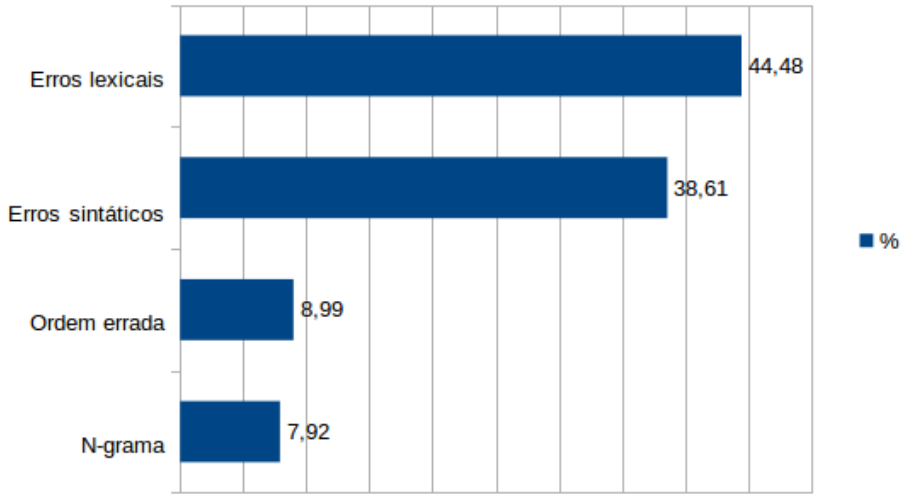


Figura 1.1: Porcentagens de ocorrência das categorias de erro anotadas no corpus de treinamento usado nesta pesquisa.

Frente à realidade atual, a principal motivação deste projeto é melhorar a qualidade da TA para o português do Brasil, por meio da investigação, implementação e avaliação de técnicas de aprendizado automático aplicadas na correção do texto traduzido automaticamente.

1.2 Objetivos

Considerando-se a grande demanda de pós-edição da TA, essa pesquisa foi realizada com o objetivo de verificar a efetividade da aplicação de um APE (pós-editor automático) na saída de

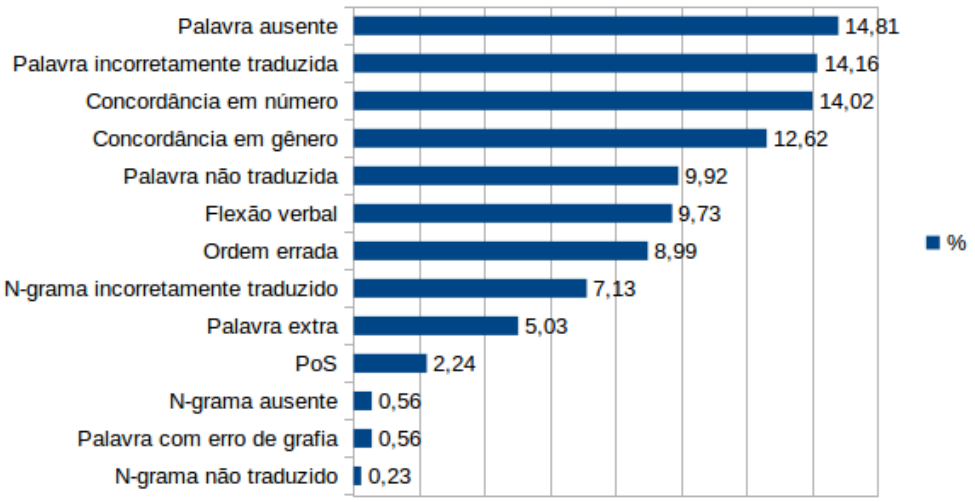


Figura 1.2: Porcentagens de ocorrência das subcategorias de erro anotadas no corpus de treinamento usado nesta pesquisa.

um tradutor automático treinado para o par de idiomas inglês-português do Brasil (*baseline*).[5] O tradutor automático usado como *baseline* nesta proposta é o tradutor automático estatístico baseado em frases (*phrase-based statistical MT*, PB-SMT) treinado usando o *toolkit* de TA Moses (KOEHN et al., 2007) e o corpus FAPESP-v2 (AZIZ; SPECIA, 2011), doravante denominado TAEIP (detalhes sobre o TAEIP podem ser obtidos na seção 5.5). Os modelos de tradução e de língua usados no TAEIP podem ser obtidos no Portal de TA, PorTAl.[6]

Para a implementação do APE optou-se por dividir a tarefa de pós-edição automática em duas etapas distintas, implementadas em módulos separados, responsáveis por: (1) identificar os erros de tradução e (2) corrigir os erros de tradução. Assim, essa pesquisa englobou: (i) a investigação, a implementação e a avaliação de técnicas de AM para a identificação de erros de TA e a correção automática de textos traduzidos automaticamente, (ii) a implementação de um APE capaz de identificar e corrigir erros de TA, principalmente aqueles cometidos pelo paradigma estatístico (erros de concordância e ordem, por exemplo), bem como (iii) a aplicação do APE em textos traduzidos automaticamente pelo TAEIP.

Desse modo, duas hipóteses de pesquisa são perseguidas neste trabalho: (H1) a pós-edição automática melhora a qualidade da (TA) e (H2) a identificação de erros realizada como passo prévio à correção evita que alterações desnecessárias sejam realizadas gerando, inclusive, novos erros.

[5] A escolha pelo idioma alvo (resultante da tradução) como o português do Brasil decorre da maior disponibilidade de falantes de tal idioma para permitir a avaliação desta proposta.

[6] Modelos para en-pt disponíveis em *Downloads*. Disponível em: <http://www.lalic.dc.ufscar.br/portal/>. Acesso em: 17 jan. 2014.

O APE resultante desta pesquisa é modular, baseado em técnicas de AM e, apesar de ter sido desenvolvido com o foco na correção de erros da TA estatística, acredita-se que possa ser usado na saída de diferentes tipos de sistemas de TA para o português do Brasil.

1.3 Organização do texto

Os próximos capítulos deste documento estão organizados como descrito a seguir.

A revisão bibliográfica referente ao tema pós-edição encontra-se no Capítulo 2. Nesse capítulo são mostrados trabalhos atuais tanto nas tarefas de categorização e anotação de erros de TA como na identificação e correção desses erros, além de ferramentas e algoritmos de aprendizado de máquina que auxiliam na implementação do APE.

O Capítulo 3 apresenta o processo de anotação do corpus de treinamento, com a definição das categorias e subcategorias de erros adotadas neste trabalho.

A metodologia adotada e os experimentos realizados para a identificação de erros e a correção de erros usando o corpus anotado conforme descrito no Capítulo 3 são mostrados nos Capítulos 4 e 5, respectivamente.

Por fim, o Capítulo 6 traz as conclusões e considerações finais desta pesquisa, bem como propostas de trabalhos futuros.

Capítulo 2

PÓS-EDIÇÃO DA TRADUÇÃO

Tradução é o processo que envolve a compreensão de um texto em uma língua fonte e a elaboração de sua representação correspondente na língua alvo desejada. A tradução pode ser executada por humanos, automaticamente, ou uma combinação de ambos. A tradução automática (TA) é o principal objeto de estudo desta pesquisa e, para familiarizar o leitor, a próxima seção (2.1) traz uma breve contextualização sobre TA, seus principais tipos e abordagens, com especial atenção para a tradução automática estatística (seção 2.1.1), o estado-da-arte, e as medidas usadas para a avaliação da saída da TA (seção 2.1.2).

Embora a TA seja menos custosa do que a tradução humana, a qualidade da saída gerada ainda está aquém do desejado em muitos casos, por isso, nos últimos anos, pesquisas têm sido desenvolvidas com o intuito de pós-editar a saída da TA para melhorar sua qualidade. Esse também é o intuito da pesquisa aqui apresentada, e como o pós-editor automático desenvolvido usa algoritmos de aprendizado de máquina (AM), este é o assunto abordado na seção 2.2. Em seguida, a seção 2.3 apresenta o processo de pós-edição da tradução automática (seção 2.3) e as duas etapas que constituem esse processo: a identificação de erros na tradução (2.3.1) e a correção de erros na tradução (2.3.2).

2.1 Tradução automática (TA)

A tradução automática (TA) ou *Machine Translation* (MT) pode ser entendida como a tarefa de traduzir textos em língua natural automaticamente. Os tipos de TA mais conhecidos são:

- **SMT (*Statistical* MT ou TA Estatística)**: utiliza medidas estatísticas a fim de determinar a tradução na língua alvo mais provável de ser a correspondente na língua fonte. Sistemas de SMT têm como base os modelos propostos por Brown et al. (1990) e podem ser construídos por *toolkits* como o Moses[1] (KOEHN et al., 2007);

[1] Disponível em: <http://www.statmt.org/moses>. Acesso em: 21 jan. 2014.

- **EBMT (*Example Based* MT ou TA Baseada em Exemplos)**: usa reconhecimento de padrões para realizar a tradução de parte de cada sentença da língua fonte em seu correspondente na língua alvo. Um exemplo de sistema de EBMT é o Pangloss (BROWN, 1996);

- **RBMT (*Rule-Based* MT ou TA baseada em Regras)**: emprega conhecimento linguístico. De acordo com a definição encontrada em (LAGARDA et al., 2009), os sistemas RBMT possuem dois componentes principais, cujos dados são gerados por linguistas: as regras sintáticas e o léxico, sendo o último composto por informação morfológica, sintática e semântica. Um exemplo de um sistema RBMT é o Apertium[2] (ARMENTANO-OLLER et al., 2006), disponível para vários pares de idiomas, entre eles o português-espanhol.

Os sistemas de TA podem ser classificados, ainda, segundo a metodologia (ou abordagem) de tradução sendo utilizada: TA direta, TA por transferência ou TA por interlíngua. De acordo com a definição encontrada em (SILVA, 2010):

- **TA direta**: usa correspondência direta entre unidades lexicais da língua fonte e da língua alvo;

- **TA por transferência**: realiza a análise sintática da frase na língua fonte e, aplicando regras de transferência sintática, monta a representação sintática na língua alvo;

- **TA por interlíngua**: utiliza uma "língua" intermediária denominada interlíngua para representar a língua fonte e a partir desta representação efetua a tradução para a língua alvo.

Além das abordagens de tradução completamente automática, outras frentes de pesquisa se desenvolveram com o intuito de criar ferramentas e recursos para auxiliar o humano na tarefa de traduzir (*Machine-Aided Human Translation*, MAHT) ou para editar a tradução antes, durante ou depois de sua realização (*Human-Aided Machine Translation*, HAMT). Nesse sentido podem ser citados os produtos da Trados[3], da IBM [4] e o Déjà Vu da Atril[5]. Um dos componentes da MAHT que podem ser aplicados para auxiliar a correção de erros é a memória de tradução (*Translation Memory*, TM). Uma memória de tradução contém partes de textos em seu formato original, ou seja, na língua fonte, e seu correspondente traduzido na língua alvo. Por meio de uma ferramenta de TM, ao realizar uma nova tradução, a sentença de entrada é comparada com

[2]Disponível em: <http://www.apertium.org/>. Acesso em: 21 jan. 2014.

[3]Disponível em: <http://www.trados.com>. Acesso em: 21 jan. 2014.

[4]Disponível em: <http://www-03.ibm.com/software/products/en/translation-server>. Acesso em: 19 fev. 2014.

[5]Disponível em: <http://www.atril.com/>. Acesso em: 21 jan. 2014.

a sentença correspondente na língua fonte armazenada na TM juntamente com seu par na língua alvo, a fim de encontrar a tradução.

Recentemente, a pesquisa em TA vivenciou uma mudança de paradigma passando do linguístico (baseado em teorias linguísticas bem definidas que especificam restrições sintáticas, lexicais ou semânticas) para o empírico (o qual utiliza pouca ou nenhuma teoria linguística no processo de tradução). Segundo Hutchins (2005), o paradigma de TA baseada em regras (*Rule-Based Machine Translation*, RBMT) dominava o cenário da TA até a década de 1980 quando as técnicas baseadas em corpus ganharam força. Essa mudança pode ser explicada, em parte, pelos avanços de *hardware* necessários para as abordagens do paradigma empírico e que não estavam disponíveis há alguns anos. Além desse fator, outro de grande relevância para tal mudança de paradigmas é a disponibilidade crescente de recursos como dicionários e corpora paralelos envolvendo várias línguas. Assim, sistemas puramente fundamentais como os RBMT têm dado espaço para EBMT e, com grande ênfase nos últimos anos, sistemas SMT apresentados a seguir.

2.1.1 Tradução automática estatística (SMT)

A tradução automática estatística ou *statistical machine translation* (SMT) faz uso de estatística e de corpus bilíngue paralelo para encontrar a probabilidade da tradução na língua alvo dado um texto na língua fonte. Em SMT, cada sentença na língua alvo é uma tradução possível da sentença fonte (BROWN et al., 1990). Desse modo, a tradução de um texto pode ser entendida como a desambiguação das possíveis traduções, buscando aquela que seja a mais adequada, ou seja, a que maximize a probabilidade de tradução. Dada uma sentença fonte f, o sistema de SMT busca a sentença que maximize a probabilidade de tradução com base nas probabilidades $P(e|f)$ atribuídas a cada tradução possível e usando o teorema de Bayes:

$$argmax_e P(e|f) = argmax_e P(f|e)P(e) \tag{2.1}$$

As duas probabilidades apresentadas na equação 2.1 representam os dois modelos construídos na tradução estatística: o modelo de língua $P(e)$ e o modelo de tradução $P(f|e)$. Enquanto o modelo de língua é usado para calcular a fluência de uma sentença gerada na língua alvo, o modelo de tradução é usado para calcular a adequação da sentença traduzida em relação à sentença original.

Um modelo de língua $P(e)$ possui palavras (ou sequências de palavras) acompanhadas da probabilidade de ocorrerem na língua alvo, obtidas a partir de um corpus monolíngue contendo textos na língua alvo. O modelo de tradução $P(f|e)$, por sua vez, possui pares de palavras (ou sequências de palavras) acompanhadas de suas probabilidades de co-ocorrência obtidas a partir de um corpus paralelo bilíngue.

O paradigma dominante na área de TA atualmente faz uso de tradução baseada em frases (OCH; NEY, 2004) também conhecido como *phrase-based* SMT (KOEHN et al., 2007).[6] Esse tipo de tradução estatística busca a frase na língua alvo mais provável de ser a tradução de uma frase da língua fonte.

Algumas das inovações da tradução baseada em frases citadas por Cancedda et al. (2009), quando comparada à abordagem estatística clássica, são: uso de modelos log-lineares[7], unidades básicas de tradução multipalavras ao invés de palavras, possibilidade do treinamento do modelo log-linear usando MERT (*Minimum Error Rate Training*) (OCH, 2003) com o objetivo de otimizar o sistema visando melhorar uma das métricas automáticas.

O sistema de TA utilizado como *baseline*[8] neste trabalho é baseado em frases e faz uso de modelos de língua e de tradução com frases de tamanhos máximos 5 (5-grama) e 7 (7-grama), respectivamente. Com relação ao tamanho máximo das frases, vale mencionar que o custo computacional cresce exponencialmente para calcular a probabilidade utilizando frases muito longas.

Além da limitação computacional na geração de modelos de frases, a TA estatística enfrenta outros problemas de cunho linguístico como a falta de concordância entre duas frases consecutivas ilustrada no exemplo a seguir:

- Sentença fonte (em inglês): "*The sedentary are going to feel* ***uncomfortable*** ,..."
- Sentença de referência (em português): "Os sedentários vão se sentir **desconfortáveis** , ..."
- Sentença traduzida pelo TAEIP (em português): "Os sedentários vão se sentir **desconfortável** , ..."
- Sentença traduzida pelo TAEIP (em português e contendo as marcações das frases[9]): "Os $|0-0|$ sedentários $|1-1|$ vão $|2-4|$ se sentir $|5-5|$ **desconfortável** $|6-6|$, $|7-7|$ ..."

Uma alternativa para melhorar a saída da TA é pós-editá-la usando um sistema de pós-edição automática, um APE (veja seção 2.3). Para verificar a efetividade do APE pode-se

[6] As frases na SMT são sequências contíguas de palavras. Em inglês elas são denominadas *phrases* embora não necessariamente representem sintagmas propriamente ditos sendo que a melhor interpretação para elas é a de n-gramas.

[7] O uso de modelos log-lineares na TA permite que contexto seja considerado na modelagem, tornando possível a definição de características que ajudem a melhorar a tradução (LOPEZ, 2008).

[8] *Baseline* é o sistema usado como base na comparação com outras estratégias propostas.

[9] Cada frase alvo está associada à frase fonte que ela traduz. Essa associação está indicada pelo intervalo de *tokens* fonte. Por exemplo, o intervalo $|0-0|$ indica que a frase alvo "Os" é a tradução da frase fonte contendo apenas o *token* na posição 0 ("The"), enquanto que o intervalo $|2-4|$ indica que a frase alvo "vão" é a tradução da frase fonte contendo os *tokens* nas posições 2 ("are"), 3 ("going") e 4 ("to").

avaliar a qualidade da tradução gerada pelo *baseline* e a de sua versão pós-editada usando medidas de avaliação automática como BLEU (PAPINENI et al., 2002) e NIST (DODDINGTON, 2002), explicadas na seção 2.1.2.

2.1.2 Medidas de avaliação da tradução automática

Conforme já mencionado, apesar de mais de meio século de pesquisas em TA, os sistemas atuais ainda apresentam desempenho aquém do desejado. Como consequência, a avaliação dos sistemas de TA desperta tanto interesse quanto a própria pesquisa por novas técnicas de TA. Avaliar manualmente a saída da tradução automática, embora seja o ideal, é uma tarefa extremamente custosa, pois necessita de especialistas humanos em ambos os idiomas usados na tradução, avaliando imensas quantidades de textos traduzidos automaticamente.

Assim sendo, paralelamente ao desenvolvimento dos sistemas de TA surgiram propostas de medidas automáticas para a avaliação da tradução produzida por esses sistemas. As medidas automáticas avaliam a qualidade de uma tradução comparando-a com uma ou mais traduções de referência produzidas por especialistas humanos e são usadas em ao menos três tarefas: (1) na avaliação absoluta, ou seja, checar se a saída TA pode ser usada em determinada aplicacão, (2) na comparação entre sistemas ou para verificar o efeito de alterações em um determinado sistema, (3) para guiar ajustes (*tuning*) nos sistemas de TA baseados em aprendizado. No caso desta pesquisa, as medidas foram usadas para a tarefa (2): comparação entre o *baseline* e o APE.

Seguindo a mesma designação proposta por Cancedda et al. (2009), as medidas baseadas em n-grama são as mais difundidas dentre as medidas automáticas atualmente. Seu cálculo considera a fração de n-gramas presentes em um conjunto de sentenças traduzidas que também estejam presentes nas suas respectivas referências (*clipped n-gram precision*). Cada n-grama na referência é usado para buscar um equivalente em, no máximo, um n-grama na saída da TA, sendo assim, a contagem é limitada (*clipped*) ao número de ocorrências de n-gramas na referência. Exemplos dessas medidas são as duas que foram selecionadas para esta pesquisa por serem as mais utilizadas na área:

- BLEU (PAPINENI et al., 2002): avalia a saída da tradução automática (C) calculando a média geométrica da precisão considerando, geralmente, N (tamanho máximo do n-grama) igual a 4. A precisão (p_n) é dada pelo número de casamentos dos n-gramas de C, sendo $w_1 ... w_n$ a representação de um n-grama em C com uma ou mais referências. Em p_n, a quantidade de casamentos entre n-gramas de C e a referência é representada por $count_{clip}(w_1 ... w_n)$ e o número de vezes em que o n-grama aparece em C é dado por $count(w_1 ... w_n)$. A esse valor é multiplicado um fator de penalidade (*brevity penalty*, ou

BP) para sentenças muito curtas em comparação com as referências, onde c é o tamanho de C e r, o tamanho médio das referências. Veja fórmulas na Figura 2.1.

Figura 2.1: Fórmulas para cálculo de BLEU.

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} \frac{1}{N} \ln p_n\right)$$

$$p_n = \frac{\sum_{w_1 \ldots w_n \in C} \text{count}_{\text{clip}}(w_1 \ldots w_n)}{\sum_{w_1 \ldots w_n \in C} \text{count}(w_1 \ldots w_n)} \qquad \text{BP} = \begin{cases} 1 & \text{se c} > \text{r} \\ \exp\left(1 - \frac{r}{c}\right) & \text{se c} \leq \text{r} \end{cases}$$

Fonte: (CASELI, 2007, p.43).

- NIST (DODDINGTON, 2002): BLEU foi o ponto de partida para sua criação. As diferenças entre o cálculo de BLEU e de NIST são as seguintes: (i) ao invés de usar a média geométrica como em BLEU, NIST usa a média aritmética e, geralmente, N igual a 5; (ii) a penalidade (*brevity penalty*, ou BP') usada também é diferente e possui o argumento β, que tem por objetivo fazer com que BP' seja 0,5 quando c é igual a $\frac{2}{3}$ de r; além disso, (iii) os n-gramas têm pesos por frequência, onde os n-gramas menos frequentes, por serem mais informativos, recebem um peso maior do que os mais frequentes, influenciando no valor de info $(w_1 \ldots w_n)$. Veja fórmulas na Figura 2.2.

Figura 2.2: Fórmulas para cálculo de NIST.

$$\text{NIST} = \text{BP}' \times \sum_{n=1}^{N} \sum_{w_1 \ldots w_n \in C} \frac{\text{info}(w_1 \ldots w_n)}{\text{count}(w_1 \ldots w_n)}$$

$$\text{info}(w_1 \ldots w_n) = \log_2 \left[\frac{\text{número de ocorrências de } w_1 \ldots w_{n-1}}{\text{número de ocorrências de } w_1 \ldots w_n}\right]$$

$$\text{BP}' = \begin{cases} 1 & \text{se c} > \text{r} \\ \exp\left(\beta \ln^2\left(\frac{c}{r}\right)\right) & \text{se c} \leq \text{r} \end{cases}$$

Fonte: (CASELI, 2007, p.44).

Nas medidas baseadas em n-grama, quanto maior o valor obtido para a medida, melhor é a TA. No caso de BLEU este valor está entre 0 e 1 ou 0 a 100 como se tem convencionado utilizar atualmente. Para NIST o valor mínimo é 0, mas não há valor máximo pré-estipulado.

Além das medidas baseadas em n-grama, Cancedda et al. (2009) apresentam outras medidas usadas na avaliação automática da TA, as quais não são abordadas neste documento porque fogem ao escopo da pesquisa aqui relatada.

2.2 Técnicas de aprendizado de máquina (AM)

As técnicas de aprendizado de máquina (AM) são utilizadas para a categorização de exemplos com base em um treinamento. Em outras palavras, essas técnicas são treinadas com base em um conjunto de exemplos acompanhados de características (*features*) de tal modo que aprendem a classificar novos exemplos a partir de seus conjuntos de características. No caso do APE desenvolvido nesta pesquisa (veja detalhes nos capítulos 4 e 5), essas técnicas foram utilizadas nas duas etapas que constituem a pós-edição automática: a identificação de erros e a correção de erros.

No AM, as técnicas são divididas de acordo com o conhecimento (ou não) das classes (ou categorias) a serem aprendidas: quando as classes a serem aprendidas são conhecidas, o aprendizado é *supervisionado* e quando se desconhecem as classes e o intuito é encontrar características que agrupem os exemplos de uma mesma classe, o aprendizado é *não-supervisionado*. Há, ainda, um terceiro tipo de aprendizado, o *semissupervisionado*, no qual apenas parte dos exemplos de treinamento está rotulada com a classe.

No aprendizado supervisionado há um conjunto de dados de treinamento rotulados com uma classe conhecida (exemplos) e o objetivo é produzir uma função que seja capaz de classificar novas instâncias com base nos exemplos (MITCHELL, 1997). Os métodos de aprendizado supervisionado que foram aplicados nos experimentos apresentados neste documento foram:

- Naive Bayes: o algoritmo Naive Bayes (JOHN; LANGLEY, 1995) busca as frequências com que a classe e cada atributo dada a classe ocorrem nos dados de treinamento e as utiliza como a hipótese aprendida. A hipótese, por sua vez, é empregada para classificar as novas instâncias, maximizando o produtório da frequência da classe pelo valor de cada atributo da nova instância dada a classe. Uma característica desse algoritmo é que ele considera independência condicional de todos os atributos dada a classe. Ao assumir independência condicional entre os atributos, a complexidade da função de aprendizado é bastante reduzida. Quando os atributos das instâncias que se deseja classificar são realmente independentes entre si tem-se uma classificação considerada ótima. No entanto, nas aplicações onde a dependência condicional é fundamental, a suposição de independência condicional é restritiva (MITCHELL, 1997). Portanto deve-se verificar com cuidado o custo-benefício caso a caso.

- Árvore de decisão: a ideia clássica de uma árvore de decisão pode ser encontrada em (QUINLAN, 1990) e a implementação da árvore C4.5, detalhada em (QUINLAN, 1993), permite a seleção ou não de poda para a execução. A árvore de decisão é bastante usada em tarefas de classificação. Usa-se o ganho de informação, uma medida baseada no

conceito de entropia[10], para criar um *ranking* dos atributos mais informativos, que por sua vez, é tomado como base para selecionar a ordem de sua inserção na árvore usada na classificação. Cada ramo da árvore é uma conjunção de testes, sendo a árvore uma disjunção de ramos (MITCHELL, 1997).

- SVM (*Support Vector Machine*): O SVM padrão faz a atribuição de uma classe entre duas possíveis (classificador binário) para determinada instância. Para isso, o algoritmo do SVM monta um modelo onde os exemplos são representados como pontos no espaço. Novos exemplos são, então, mapeados nesse espaço e recebem a classe de acordo com o lado do espaço ao qual foram atribuídos. A implementação usada nesse trabalho é a de Platt e os problemas multiclasse são solucionados usando a implementação de Hastie e Tibshirani.

- TBL (*Transformation-based Learning*) (BRILL, 1995): De acordo com a definição encontrada em (ELMING, 2006), essa técnica busca qual regra de correção causa a maior redução de erro em um determinado passo do treinamento. Depois disso, aplica a regra encontrada aos dados de treinamento e volta a buscar e aplicar a melhor regra nos dados corrigidos pela anterior, e assim por diante até que a diminuição dos erros atinja um nível abaixo do definido previamente. Finalmente tem-se uma lista de regras e a prioridade na qual devem ser aplicadas com o objetivo de reduzir os erros ao máximo. Embora essa técnica não tenha classes associadas às instâncias, ela pode ser considerada supervisionada porque cada instância é formada pelo exemplo incorreto e o exemplo correto de tal modo que ela aprende como transformar o incorreto no correto. Como apontado por Lager (1999), essa técnica de AM é usada para aprender regras para muitas aplicações na área de PLN, tais como etiquetação morfossintática (BRILL, 1995) e correção de erros de grafia (MANGU; BRILL, 1997), dentre outras.

Os processos de avaliação de modelos treinados através dos algoritmos de AM são geralmente baseados no cálculo da precisão e cobertura. A medida da precisão representa a proporção de instâncias que realmente pertençam a uma determinada classe x, dentre todas aquelas que foram classificadas, correta ou incorretamente, como pertencentes à classe x. Já a cobertura representa a proporção de instâncias que foram classificadas como pertencentes à classe x, dentre todas as instâncias que realmente pertençam à classe x. Alguns autores também usam uma medida que combina precisão e cobertura, chamada de medida-F, que é calculada como $\frac{2*precisao*cobertura}{(precisao+cobertura)}$ (HALL et al., 2009).

No momento de treinar um modelo usando um algoritmo de AM pode-se utilizar apenas

[10] A entropia (E) é dada pela fórmula $E(T) = -\sum_{i=1}^{|C|} P_T(c_i) log_2 P_T(c_i)$, sendo $|C|$ o número de classes de um conjunto de treinamento T, c_i um valor de C e $P_T(c_i)$ a porcentagem de amostras que têm o valor c_i atribuído em T (SANTOS, 2009).

um conjunto de instâncias (com classes já atribuídas) para treinamento e teste ou, mais recomendável, conjuntos diferentes: um deles para o treinamento do modelo e outro para teste. Uma outra alternativa para testar um modelo de AM é a validação cruzada ou *cross-validation*. Na validação cruzada o conjunto de instâncias é aleatoriamente dividido em n conjuntos de igual tamanho. São feitas n iterações e, a cada uma delas, um dos conjuntos é aplicado na avaliação de um modelo que foi treinado usando os $n-1$ conjuntos restantes. O resultado final da validação cruzada é dado, então, como a média dos resultados obtidos por todos os conjuntos após n iterações de treinamento-teste.

Quando um modelo treinado apresenta um alto desempenho, em termos de precisão e cobertura, ao ser avaliado nas instâncias do conjunto de treinamento, porém apresenta um baixo desempenho em um conjunto de teste diferente do conjunto de treinamento diz-se que houve uma superespecialização ou *overfitting*. A fim de evitar o *overfitting*, e sempre que possível, é recomendada a utilização de um conjunto de instâncias para teste que seja diferente das instâncias usadas no treinamento. Um fator que pode levar ao *overfitting* é um conjunto de treinamento pequeno ou com pouca representatividade do problema que se deseja aprender.

2.3 Pós-edição da tradução automática

O processo de pós-edição da tradução automática é normalmente executado por um tradutor humano que faz modificações no texto de saída da TA com o intuito de transformá-lo em um texto aceitável para o propósito desejado (KRINGS, 2001). Para executar tal tarefa, o tradutor humano faz uma comparação do texto fonte (original) com a saída do tradutor automático (tradução). Assim, enquanto o processo de tradução geralmente tem como entrada apenas um texto escrito na língua fonte a ser transformado na língua alvo, o de pós-edição tem como entrada o texto no idioma original e a saída da TA. Texto original e traduzido automaticamente são, ambos, utilizados pelo tradutor para gerar o texto final que, dependendo do seu uso, precisará ser produzido em um padrão publicável ou apenas deverá estar em um padrão que permita um entendimento geral (O'BRIEN, 2002).

Deve ficar claro, portanto, que a pós-edição efetuada por humanos é diferente da revisão ou tradução tradicionais. Ao revisar ou traduzir um texto o tradutor trabalha para que o texto produzido fique o mais próximo possível da estrutura comumente utilizada na língua alvo. Já na pós-edição basta que o texto final esteja de acordo com o significado original e as regras básicas da língua alvo, ainda que siga mais fielmente a estrutura do texto fonte.

Os erros encontrados por um pós-editor também diferem dos erros tradicionais das traduções geradas manualmente, já que inconsistências nos textos produzidos por um tradutor humano e um tradutor automático geralmente são diferentes na frequência, repetitividade e

tipos. Por exemplo, um tradutor humano pode traduzir mal uma palavra ou estrutura no texto apenas uma vez, enquanto é bem provável que um tradutor automático erre repetidas vezes ao traduzir a mesma palavra ou tipo de estrutura (KRINGS, 2001).

Embora ainda seja uma tarefa tradicionalmente executada por humanos, há várias pesquisas em andamento e ferramentas já criadas para auxiliar e/ou automatizar a tarefa de pós-edição da tradução automática. A seguir, são apresentadas separadamente as duas etapas principais da pós-edição, acompanhadas do relato de trabalhos na literatura: identificação de erros na tradução (seção 2.3.1) e correção de erros na tradução (seção 2.3.2). É importante mencionar que, na pós-edição automática, a etapa de identificação de erros é opcional sendo possível partir direto para a correção propriamente dita. Contudo, uma das hipóteses desta pesquisa é que a identificação de erros previne a inserção de novos erros uma vez que evita a alteração desnecessária de trechos corretos na tradução.

2.3.1 Identificação de erros na tradução

Uma parte importante da pós-edição é a fase de avaliação da qualidade da TA para se determinar quais trechos precisam de correção. Essa avaliação, quando executada antes da correção, evita que uma determinada saída da TA, de boa qualidade ou que não apresente erros do tipo que se deseja tratar, seja desnecessariamente pós-editada. A avaliação da qualidade da TA pode ser feita, por exemplo, dando-se notas a segmentos traduzidos ou por meio da identificação de erros com base em categorias pré-definidas. O processo de verificação de sentenças traduzidas pode ser feito manualmente ou automaticamente.

A seguir são apresentadas diferentes formas manuais (subseção 2.3.1.1) ou automáticas (subseção 2.3.1.2) de avaliação da tradução e anotação dos erros. Em alguns dos trabalhos consultados são usadas categorias linguisticamente motivadas, outros autores fazem uso de alinhamento da saída da TA com tradução(ões) de referência, e outros aprendizado de máquina. Ao final desta seção é apresentado um resumo das principais características dos trabalhos mencionados (subseção 2.3.1.3).

2.3.1.1 Identificação manual de erros

Os trabalhos de anotação manual dos erros na tradução propõem categorias para serem utilizadas por avaliadores humanos. Em (VILAR et al., 2006), os autores organizam as categorias de erro em uma tipologia composta por vários níveis. O nível mais alto da tipologia contém cinco classes de erro: (1) palavra ausente, (2) ordem de palavra, (3) palavra incorreta, (4) palavra não conhecida e (5) pontuação. Com exceção da classe pontuação, todas as classes contêm vários subníveis. Para Vilar et al. (2006), a criação das categorias foi motivada pela

necessidade de identificar com mais precisão os problemas de um tradutor automático, pois os valores gerados pelas medidas mais comumente usadas na avaliação da TA – WER[11], PER[12], BLEU e NIST (veja seção 2.1.2) –, não são facilmente relacionados com os erros da saída da TA.

Outro estudo de avaliação por humanos pode ser encontrado em (CALLISON-BURCH et al., 2007) onde são verificadas escalas distintas para fluência (quão fluente é a tradução) e adequação (quanto a sentença traduzida consegue expressar o que está na referência). A escala de fluência varia de 1 (incompreensível) a 5 (sem falhas) e a de adequação vai de 1 (nenhuma) a 5 (total). Ambas as escalas foram desenvolvidas para o NIST *Machine Translation Evaluation Workshop by the Linguistics Data Consortium* de 2005.

Em (FARRÚS et al., 2010), propõem-se categorias de erros linguisticamente motivadas para que sejam usadas como complemento às avaliações automáticas. As categorias propostas são divididas em níveis linguísticos: ortográfico (por exemplo, pontuação e acento), morfológico (por exemplo, concordância nominal e verbal), lexical (palavras incorretamente traduzidas, não traduzidas, entre outros), semântico (polissemia, homonímia, expressões incorretas) e sintático (por exemplo, artigo ausente ou extra, erros em preposições). Concluiu-se, no referido estudo, que a avaliação humana dos níveis lexical e semântico são aparentemente consistentes com as medidas BLEU e TER[13] para as quais foi feita essa comparação.

Ferramenta para anotação manual de erros de tradução

Para auxiliar o humano na tarefa de anotação de erros existem algumas ferramentas, entre elas a Blast[14] (STYMNE, 2011a). Uma das vantagens do uso da Blast, utilizada nos estudos linguísticos descritos no capítulo 3, é o fato de possuir algumas tipologias de erro facilmente adaptáveis. Outros pontos fortes da ferramenta são sua agilidade na anotação de erros e o fato de permitir a geração de estatísticas referentes aos erros anotados. Outra característica da Blast é a gravação dos arquivos anotados em um padrão que pode ser facilmente convertido em tabelas para posterior uso como parâmetros de entrada em experimentos. Os arquivos de entrada e saída são como se segue:

- Entrada – três arquivos contendo:

[11] WER (*Word Error Rate*) (NIEßEN et al., 2000): é calculada através da normalização – usando o comprimento da sentença de referência – da soma das substituições, exclusões ou inserções.

[12] PER (*Position-independent word Error Rate*) (TILLMANN et al., 1997): é calculada com base no tamanho da intersecção entre as *bag of words* (conjunto de palavras representadas como uma "sacola" de palavras, ou seja, sem levar em consideração a ordem ou a gramática) da tradução e da referência, e depois normaliza-o pelo tamanho da referência.

[13] TER (*Translation Edit Rate*) (SNOVER et al., 2006): é calculada de modo similar à WER, pois conta o número mínimo de inserções, exclusões e substituições, mas além disso considera a mudança de uma sequência de palavras (frase) de um local para outro da sentença, operação chamada de *shift*.

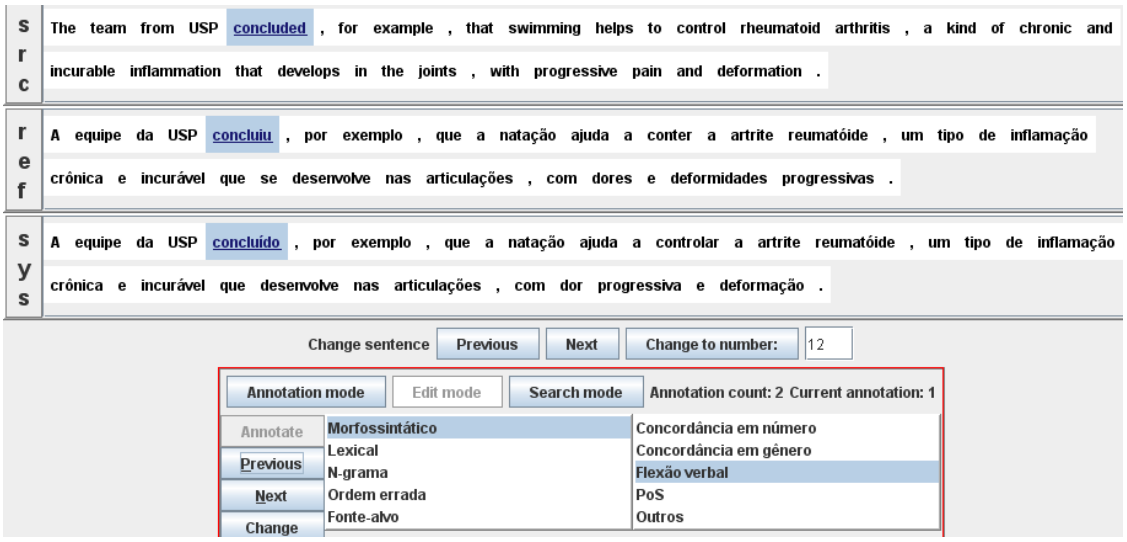
[14] Disponível em: <http://www.ida.liu.se/∼sarst/blast/>. Acesso em: 21 jan. 2014.

  - sentença(s) fonte (src)
  - sentença(s) de referência (ref)
  - sentença(s) traduzida(s) automaticamente (sys)

- Saída (exemplo na Figura 2.5) – um arquivo contendo blocos de:
  - sentença fonte
  - sentença de referência
  - sentença traduzida
  - posição(ões) da(s) palavra(s) com erro na sentença fonte
  - posição(ões) da(s) palavra(s) com erro na sentença de referência
  - posição(ões) da(s) palavra(s) com erro na sentença traduzida
  - código do erro

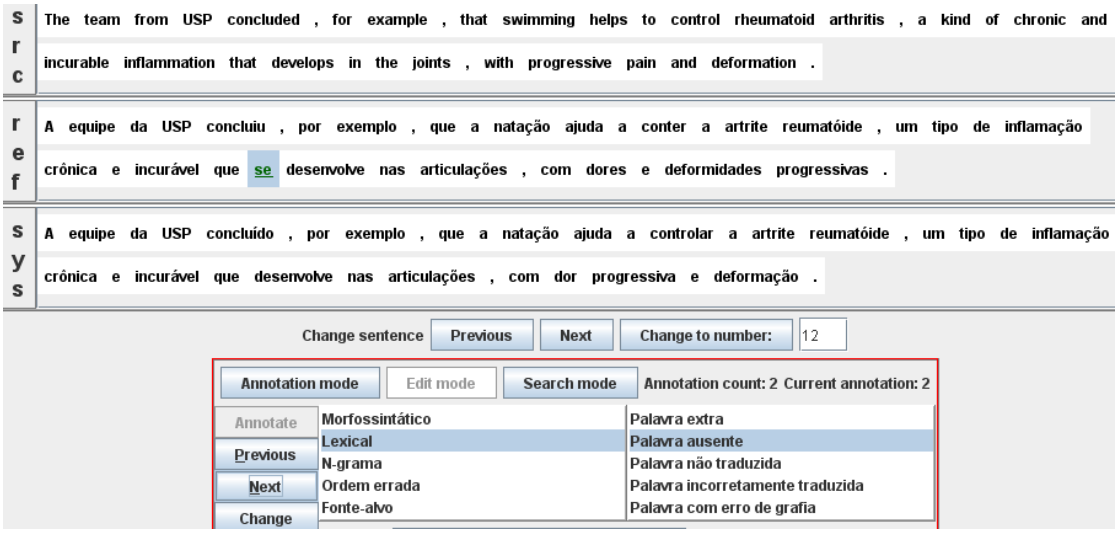
A Figura 2.3 traz um exemplo de anotação de erro de flexão verbal no qual são anotadas: a palavra original na sentença fonte (src), a palavra correta na sentença de referência (ref) e a tradução com erro de flexão verbal na saída do TA (sys). A Figura 2.4, por sua vez, apresenta a anotação de um erro de palavra ausente, o qual só é anotado na sentença de referência (ref), pois não há nenhuma ocorrência correspondente para ela na sentença fonte (src) nem na sentença traduzida automaticamente (sys).

Figura 2.3: Sentença sendo anotada manualmente com o auxílio da ferramenta Blast. Exemplo de anotação para erro de flexão verbal.



A Figura 2.5 mostra o trecho do arquivo de saída gerado pela Blast contendo o bloco para os erros anotados nas Figuras 2.3 e 2.4. Ambos os erros são apresentados na última linha da

Figura 2.4: Sentença sendo anotada manualmente com o auxílio da ferramenta Blast. Exemplo de anotação para erro de palavra ausente.



figura, separados por um espaço em branco. Note que quando nenhuma palavra é anotada na sentença fonte ou de referência o valor da posição é preenchido com -1 como ocorre para o segundo erro.

Figura 2.5: Trecho do arquivo de saída da Blast para as anotações mostradas nas Figuras 2.3 e 2.4.

```
The team from USP concluded , for example , that swimming helps to control rheumatoid arthritis , a kind of chronic and incurable inflammation that develops in the joints , with progressive pain and deformation .
A equipe da USP concluiu , por exemplo , que a natação ajuda a conter a artrite reumatóide , um tipo de inflamação crônica e incurável que se desenvolve nas articulações , com dores e deformidades progressivas .
A equipe da USP concluído , por exemplo , que a natação ajuda a controlar a artrite reumatóide , um tipo de inflamação crônica e incurável que desenvolve nas articulações , com dor progressiva e deformação .

4#4#4#morph-verbFlex -1#-1#27#lex-abstWord
```

Além da Blast, que foi a ferramenta escolhida para ser utilizada nesta pesquisa pelas vantagens já citadas, outras ferramentas foram desenvolvidas com o propósito de auxiliar a pós-edição (GOMES; PARDO, 2008; KAWAMORITA; CASELI, 2012) ou avaliar a saída da TA (AZIZ; SOUSA; SPECIA, 2012), entre outras.

2.3.1.2 Identificação automática de erros

Os trabalhos encontrados na literatura que realizam a identificação automática de erros na tradução podem ser divididos de acordo com a abordagem em:

- Identificação automática de erros com base no uso da tradução de referência: (POPOVIC, 2011), testado em (POPOVIC; BURCHARDT, 2011), e (ZEMAN et al., 2011);

- Identificação automática de erros com o uso de algoritmos de AM: (FISHEL et al., 2012) e (FELICE; SPECIA, 2012).

Identificação automática com base no uso da tradução de referência

A partir de textos traduzidos automaticamente e suas traduções de referência, acompanhadas obrigatoriamente das respectivas formas base e opcionalmente de etiquetas morfossintáticas das palavras, a ferramenta Hjerson[15] (POPOVIC, 2011) classifica os erros de tradução em: morfológicos, de ordenação, palavras ausentes, palavras extras e lexicais. A ferramenta também fornece a contagem e a taxa de cada categoria de erro, por sentença ou por documento, e etiqueta a sentença de referência e a saída da TA com as classes de erro detectadas. A Figura 2.6 mostra uma visão geral do sistema.

O método implementa o algoritmo de distância de edição (LEVENSHTEIN, 1966), fazendo a identificação de cada palavra que faz parte dos erros do tipo WER e dos erros introduzidos por Popovic e Ney (2007) do tipo PER (TILLMANN et al., 1997) baseados em cobertura/precisão: HPER[16] e RPER[17].

Popovic (2011) cita que a ferramenta obteve uma boa correlação com o julgamento humano. Em (POPOVIC; BURCHARDT, 2011), as categorias de erro escolhidas para os testes foram: erros de flexão, erros de ordem, palavras ausentes, palavras extras e palavras incorretamente traduzidas. As comparações das marcações das categorias na identificação automática, realizada pela ferramenta Hjerson, com as marcações das categorias na identificação manual mostraram bons resultados atingindo coeficientes de correlação *Spearman* acima de 0,7.

Na ferramenta Addicter (*Automatic Detection and DIsplay of Common Translation ERrors*)[18] (ZEMAN et al., 2011), por sua vez, os erros são identificados levando-se em conta o alinhamento entre a hipótese (saída do sistema de TA) e a referência. Para os erros de ordenação também é usado um grafo direcionado com pesos nos nós. As categorias foram adaptadas das definidas em (VILAR et al., 2006), mas algumas pertencentes a níveis mais genéricos não foram incluídas.

Tanto Hjerson quando Addicter, segundo seus autores, são independentes de língua. Porém, ambas as ferramentas utilizam informações linguísticas específicas dos idiomas processados e, por isso, necessitam que ferramentas (como lematizadores e *taggers*) específicas dos idiomas em questão sejam aplicadas previamente.

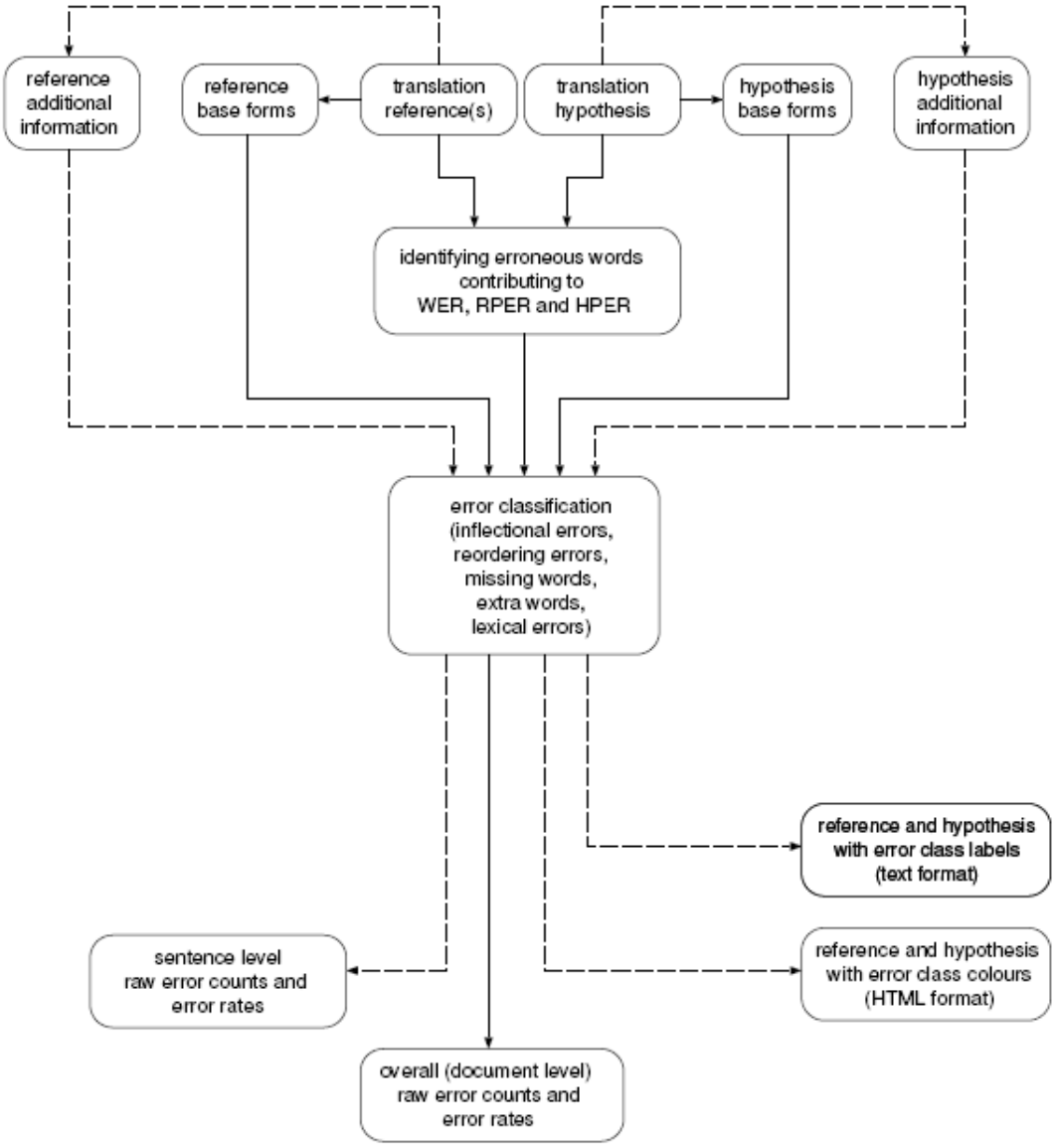
[15] Hjerson. Disponível em: <http://www.dfki.de/∼mapo02/hjerson/>. Acesso em: 21 jan. 2014.

[16] *Hypothesis* PER – casos de edição para as palavras que estão na saída da tradução automática, mas não estão na tradução de referência. No caso de várias traduções de referência, a que apresentar o menor valor de WER é a escolhida.

[17] *Reference* PER – casos de edição para as palavras que estão na referência, mas não estão na saída da tradução automática. No caso de várias traduções de referência, a que apresentar o menor valor de WER é a escolhida.

[18] Disponível em: <https://wiki.ufal.ms.mff.cuni.cz/user:zeman:addicter>. Acesso em: 21 jan. 2014.

Figura 2.6: Fluxo de processamento da ferramenta Hjerson para classificação automática de erro na qual as linhas contínuas representam entradas e saídas obrigatórias e as tracejadas, dados opcionais



Fonte: (POPOVIC, 2011, p.61).

Os resultados dos experimentos com Addicter apresentaram uma taxa de cobertura (número de instâncias classificadas) alta, mas uma precisão (número de instâncias corretamente classificadas) baixa, sendo que a precisão da categoria de erro para palavra ausente foi a mais baixa como mostrado na Tabela 2.1. Para melhorar o desempenho da ferramenta, Zeman et al. (2011) pretendem incluir suporte total à lematização para lidar melhor com línguas que tenham muita flexão. Outro desenvolvimento planejado é a inclusão da análise baseada em estrutura ou em frase (*phrase*).

[19]Para Vilar et al. (2006), uma palavra do tipo "conteúdo" é aquela que é essencial para se determinar o significado de uma sentença. Já uma palavra do tipo "auxiliar", por exemplo, uma preposição, é aquela que não altera o sentido da sentença se não estiver presente.

Tabela 2.1: Precisão e cobertura de cada grupo de erro combinando o alinhador do Addicter a outros dois alinhadores externos.

Palavra inconsistente			**Palavras fora de ordem**		
Categoria	**Precisão**	**Cobertura**	**Categoria**	**Precisão**	**Cobertura**
extra	19,24	64,68	pares	14,42	48,88
não traduzida	13,39	12,98	várias	2,47	47,69
forma	38,16	40,62			
incorreta	18,48	75,91			
Palavra da referência ausente[19]			**Erro de pontuação**		
conteúdo	2,17	15,28	pontuação	29,75	81,65
auxiliar	4,78	27,23			

Fonte: Adaptado de (ZEMAN et al., 2011).

Identificação automática com o uso de algoritmos de AM

Seguindo uma abordagem diferente, nos trabalhos que utilizam algoritmos de AM, a identificação de erros é obtida como co-produto da avaliação da TA. Nesses casos, sabe-se que uma tradução está errada porque uma baixa pontuação é atribuída a ela. Por exemplo, a ferramenta TerrorCat[20] (FISHEL et al., 2012) gera um *ranking* de traduções usando como entrada várias traduções feitas por sistemas de TA diferentes para um mesmo texto fonte e algoritmos de AM. Para tanto, Fishel et al. (2012) propõem avaliar a qualidade da saída da TA com base na comparação par a par de cada possível hipótese de tradução. Nesse método, cada categoria de erro tem pesos diferentes quando se avalia a qualidade da TA como um todo. Por exemplo, um erro de pontuação pode ser tomado como tendo menos influência do que uma palavra ausente ou determinada categoria de erro pode ter um efeito mais degradante em um idioma do que em outro. Uma nota normalizada é atribuída a cada hipótese tendo como base as frequências de categorias de erros. A categorização de erros de tradução é obtida através de duas ferramentas: Addicter (ZEMAN et al., 2011) e Hjerson (POPOVIC, 2011) descritas previamente. Além disso, TerrorCat precisa que o texto passe por lematização e etiquetação morfossintática.

O método calcula as frequências das categorias de erro (obtidas automaticamente usando as ferramentas já citadas) de cada hipótese da tradução sentencialmente e as representa par a par usando um vetor de características, onde as características obtidas dos dois sistemas são incluídas lado a lado. Depois, essas características são passadas a um classificador SVM binário que classifica a melhor sentença de cada par. A nota final dada a um sistema é calculada com base no total de sentenças melhores obtidas por ele. Já a nota de um sistema individualmente é a média das notas de suas sentenças. O classificador SVM binário é treinado usando o corpus do WMT[21], de anos anteriores ('07-'11) (contém entradas de duas a cinco sentenças geradas por diferentes sistemas de TA avaliadas comparativamente por humanos) convertido em todos

[20]Disponível em: <https://github.com/fishel/TerrorCat/>. Acesso em: 21 jan. 2014.

[21]Mais informações sobre o mais recente *Workshop on Statistical Machine Translation* (WMT) e WMT's de anos anteriores podem ser obtidas em <http://www.statmt.org/wmt13/>. Acesso em: 22 jan. 2014.

os possíveis pares de sentenças.

Usando o corpus do WMT'11 para teste, os sistemas de TA na tradução para o inglês obtiveram uma concordância de 0,86 com os julgamentos humanos e 0,85 do inglês para outras línguas avaliando o sistema de TA como um todo. Já no nível sentencial, a concordância do ranking gerado pelo TerrorCat com a avaliação feita por humanos foi bem mais baixa: 0,27 para o inglês e 0,23 do inglês para outras línguas. A comparação foi feita usando o coeficiente de *Spearman*.

Outro sistema que usa AM é o WLV-SHEF, proposto por Felice e Specia (2012). WLV-SHEF emprega características linguísticas para estimar a qualidade da saída da TA. No total foram utilizadas 70 características (*features*) linguísticas como: (a) porcentagem de verbos, pronomes e substantivos e a razão entre a entrada da TA e a saída da TA e (b) largura e profundidade de árvores de constituintes da entrada da TA e saída da TA e suas diferenças. Além de características unicamente linguísticas, há uma versão alternativa do sistema que emprega 77 características adicionais não-linguísticas, como: (a) razão entre entrada e saída da TA para comprimento de sentenças e (b) frequência média de um *token*. O conjunto de treinamento, além das traduções, trazia uma média da nota de 1 a 5 para esforço de pós-edição dada previamente por três humanos. Para modelar a tarefa foi usado SVM.

Na fase de testes, onde o WLV-SHEF é comparado com um sistema *baseline* contendo 17 características superficiais, não houve melhora, inclusive o desempenho do WLV-SHEF foi pior. Contudo a piora não foi estatisticamente significativa, o que demonstra que talvez para o conjunto de testes as características embutidas ao *baseline* já fossem suficientes para cumprir o propósito de estimar a qualidade (FELICE; SPECIA, 2012). Com base nesses resultados, Felice e Specia (2012) apontam como trabalho futuro a possibilidade de investigar a aplicação de novas características linguísticas no modelo.

2.3.1.3 Resumo dos trabalhos de identificação de erros

As Tabelas 2.2, 2.3 e 2.4 mostram os trabalhos relacionados à avaliação de textos traduzidos por TA e/ou identificação automática do que seja um erro.[22]

[22]Os idiomas citados nas tabelas são es: espanhol, en: inglês, ch: chinês, de: alemão, fr: francês, tc: tcheco, ca: catalão, ar: árabe.

[23]Disponível em: <http://www.project-syndicate.com/>. Acesso em: 15 out. 2012.

[24]GALE – *Global Autonomous Language Exploitation*. Disponível em: <http://www.arpa.mil/ipto/programs/gale/index.htm>. Acesso em: 15 out. 2012.

Tabela 2.2: Resumo dos trabalhos referentes à avaliação e/ou detecção manual de erros.

Referência	Idiomas	Saída da TA	Corpus	Categorias	Abordagem
(VILAR et al., 2006)	es-en, en-es, ch-en	SMT	es-en e en-es: discursos de sessões do parlamento europeu, ch-en: notícias fornecidas pelo LDC (*Linguistic Data Consortium*)	Nível mais alto contém erros de: palavra ausente, ordem de palavra, palavras incorretas, palavras não conhecidas e pontuação. Mais detalhes e subníveis em (VILAR et al., 2006)	Propõe categorias para serem utilizadas por avaliadores humanos
(CALLISON-BURCH et al., 2007)	en-de, de-en, en-es, es-en, en-fr, fr-en, en-tc, tc-en	SMT	Corpus *Europarl* e corpus *Project Syndicate*[23]	Escalas distintas para fluência e adequação de uma sentença traduzida variando de 1 (incompreensível) a 5 (sem falhas) e nota de 1 (nenhuma) a 5 (total) comparando cinco sentenças traduzidas por diferentes sistemas de TA, além de variação da anterior, analisando os constituintes das sentenças passadas por um *parser* e automaticamente alinhadas com as traduções de referência. Mais detalhes em (CALLISON-BURCH et al., 2007)	Avaliação feita por humanos e posteriormente comparada com várias medidas de avaliação automáticas
(FARRÚS et al., 2010)	es-ca, ca-es	SMT	es: 711 sentenças dos jornais *El País* e *La Vanguardia*, ca: 813 sentenças do jornal *Avui* e transcrições do programa de TV *Àgora*	Divididas em níveis linguísticos: ortográfico, morfológico, lexical, semântico e sintático. Detalhadas em (FARRÚS et al., 2010)	Categorias usadas somente para identificação feita por humanos

Tabela 2.3: Resumo dos trabalhos referentes à avaliação e/ou detecção automática de erros que se baseiam no alinhamento com a referência.

Referência	Idiomas	Saída da TA	Corpus	Categorias	Abordagem
(POPOVIC, 2011)	ar-en, ch-en, de-en	SMT	Traduções para o inglês realizadas por sistemas de tradução do projeto GALE[24]	Principais categorias de (VILAR et al., 2006)	Usa o algoritmo de distância de edição (LEVENSHTEIN, 1966) e obtém a identificação das palavras que contribuem para os erros do tipo WER e os do tipo RPER e HPER.
(ZEMAN et al., 2011)	en-tc	SMT	Traduções de notícias do WMT'09	Adaptadas de (VILAR et al., 2006)	Baseia-se no alinhamento da referência-texto traduzido por TA

2.3.2 Correção de erros na tradução

Assim como ocorre na etapa de identificação dos erros de tradução, a correção dos erros também pode seguir diferentes abordagens, das quais destacam-se as abordagens automáticas

Tabela 2.4: Resumo dos trabalhos referentes à avaliação e/ou detecção automática de erros que usam algoritmos de aprendizado de máquina.

Referência	Idiomas	Saída da TA	Corpus	Categorias	Abordagem
(FISHEL et al., 2012)	fr-en, de-en, es-en, tc-en, en-fr, en-de, en-es, en-tc	SMT	*Ranking* de avaliações manuais dos anos 2007-2011 do WMT (*Workshop on Machine Translation*)	(POPOVIC; NEY, 2011),(ZEMAN et al., 2011)	SVM
(FELICE; SPECIA, 2012)	en-es	SMT	Textos jornalísticos	Notas de 1 a 5 relacionadas ao esforço de pós-edição	SVM

listadas a seguir e detalhadas nas próximas subseções:

- Correção automática baseada em SMT;
- Correção automática auxiliada por memórias de tradução;
- Correção automática com o uso de verificadores gramaticais;
- Correção automática com o uso de algoritmos de AM.

Entre as abordagens citadas acima não estão incluídos os trabalhos que realizam a correção da TA usando regras criadas manualmente como (AVANÇO; NUNES, 2013). As regras manuais foram criadas tomando como base o corpus teste-a (AZIZ; SPECIA, 2011) anotado manualmente com erros de TA (mesmo corpus usado neste trabalho e apresentado em detalhes no Capítulo 3). As regras foram criadas para corrigir erros de concordância nominal e verbal, em traduções de inglês para português do Brasil, para a saída do TAEIP (mesmo sistema usado como *baseline* neste trabalho, descrito na seção 5.5). Quando as regras manuais foram aplicadas a erros anotados manualmente houve um ganho de 2,85% em BLEU e 0,99% em NIST na avaliação de 35 sentenças para concordância nominal; e de 0,46% em BLEU e 0,29% em NIST na avaliação de 40 sentenças para concordância verbal.

2.3.2.1 Correção automática baseada em SMT

Uma das abordagens utilizadas por sistemas de pós-edição automática da TA é a realizada por meio de um sistema de tradução estatística (SMT). Exemplos de trabalhos nos quais a saída da TA é pós-editada usando um tradutor SMT monolíngue são: (BÉCHARA; MA; GENABITH, 2011), (POTET et al., 2011), (UENISHI, 2013), (LAGARDA et al., 2009) e (SIMARD; GOUTTE; ISABELLE, 2007). Nos três primeiros, a TA sendo pós-processada é do tipo SMT e nos dois últimos a TA que passa por pós-edição é do tipo RBMT.

Em (BÉCHARA; MA; GENABITH, 2011) usa-se um tradutor automático do tipo SMT treinado com a própria saída da TA e sua correspondente referência para criar um tradutor monolíngue SMT a ser usado como pós-editor. Essa primeira variação do pós-editor é chamada de PE. Em uma segunda variação, chamada de PE-CF, ou pós-edição contextual, a frase ou palavra da saída da TA a ser usada para treinamento do tradutor SMT que será usado como pós-editor é concatenada com sua correspondente na língua fonte. Há, ainda, variações da versão PE-CF, as quais utilizam as pontuações mínimas do alinhamento lexical entre a sentença fonte e a traduzida automaticamente (determinado por GIZA++) para decidir quais palavras devem ser concatenadas: PE-C06, PE-C07, PE-C08 e PE-C09 para pontuações mínimas de alinhamento 0,6, 0,7, 0,8 e 0,9, respectivamente.

As Tabelas 2.5 e 2.6 mostram quantas sentenças obtiveram valores de BLEU (PAPINENI et al., 2002) melhores, piores ou iguais após a pós-edição considerando-se as duas direções de tradução (francês-inglês e inglês-francês, respectivamente) nas diferentes configurações dos sistemas de pós-edição. Embora a pós-edição PE tenha apresentado pouca melhora na saída da TA, a combinação de PE com o uso de contexto (PE-CF) e pontuação mínima de alinhamento (PE-C0n) gerou melhorias estatisticamente relevantes na tradução de dados técnicos.

Tabela 2.5: Quantidade de sentenças que apresentaram melhora, piora ou nenhuma alteração nos valores de BLEU para a tradução francês-inglês no APE desenvolvido por Béchara, Ma e Genabith (2011).

Sistema	Melhor	Pior	Igual
PE	137	88	1742
PE-CF	489	511	967
PE-C06	511	451	1005
PE-C07	497	460	1010
PE-C08	528	455	930
PE-C09	496	454	1017

Fonte: Adaptado de (BÉCHARA; MA; GENABITH, 2011).

Tabela 2.6: Quantidade de sentenças que apresentaram melhora, piora ou nenhuma alteração nos valores de BLEU para a tradução inglês-francês no APE desenvolvido por Béchara, Ma e Genabith (2011).

Sistema	Melhor	Pior	Igual
PE	166	179	1662
PE-CF	259	434	1274
PE-C06	198	318	1451
PE-C07	238	317	1412
PE-C08	181	265	1531
PE-C09	170	306	1491

Fonte: Adaptado de (BÉCHARA; MA; GENABITH, 2011).

Seguindo a mesma estratégia de (BÉCHARA; MA; GENABITH, 2011), Uenishi (2013) realizou

o treinamento de um APE estatístico para o TAEIP (sistema PB-SMT *baseline* desta pesquisa). Para tanto, o corpus utilizado no treinamento do APE estatístico foi formado pelos textos em português traduzidos automaticamente pelo TAEIP e os textos originais (gerados por humanos) em português presentes no corpus FAPESP-v2 (AZIZ; SPECIA, 2011). As mesmas ferramentas e os mesmos parâmetros usados no treinamento do TAEIP foram empregrados no treinamento do APE estatístico. Após o treinamento, o APE gerado foi aplicado para pós-processar as saídas do TAEIP em dois corpora de teste: teste-a e teste-b (AZIZ; SPECIA, 2011). Os resultados, em termos de BLEU e NIST, podem ser observados na Tabela 2.7.

Tabela 2.7: Avaliação do APE estatístico desenvolvido por Uenishi (2013) comparado à saída do TAEIP.

	TAEIP		**APE**	
	teste-a	**teste-b**	**teste-a**	**teste-b**
BLEU	59,26	48,35	56,45	46,42

Fonte: Adaptado de (UENISHI, 2013)

Como em (BÉCHARA; MA; GENABITH, 2011), Uenishi (2013) também realizou uma análise de desempenho para cada sentença dos corpora de teste. A Tabela 2.8 apresenta o desempenho comparado do APE e da TA, sentença a sentença, apontando a quantidade de sentenças para as quais os valores de BLEU foram melhores, piores ou iguais.

Tabela 2.8: Quantidade de sentenças que apresentaram melhora, piora ou nenhuma alteração nos valores de BLEU para a tradução inglês-português no APE estatístico desenvolvido por Uenishi (2013).

Corpus	**Melhor**	**Pior**	**Igual**
Teste-a	194	457	663
Teste-b	270	526	652

Fonte: Adaptado de (UENISHI, 2013)

Como constatado pelos valores das Tabelas 2.7 e 2.8, o uso do APE estatístico de (UENISHI, 2013) piorou a qualidade das sentenças traduzidas pelo TAEIP. Essa constatação foi confirmada pela análise manual de 165 trechos diferentes nas sentenças traduzidas pelo TAEIP e pós-editadas pelo APE estatístico. A partir dessa análise observou-se que 41,2% dos trechos foram melhor traduzidos pelo TAEIP, enquanto apenas 20,6% foram melhorados pelo APE.

Já Potet et al. (2011) propõem um método interativo onde a análise humana é integrada ao tradutor automático estatístico que é treinado novamente empregando a saída da TA e a mesma saída pós-editada manualmente. Os autores acreditam que sistemas de TA podem aprender ao receberem retorno dos usuários através de pós-edição. Em (POTET et al., 2011) são apresentados três métodos distintos para re-treinamento do tradutor usando pós-edições humanas:

- **Incorporação da saída da TA pós-editada ao corpus de treinamento do sistema de TA**: ao incluir a saída da TA pós-editada ao corpus de treinamento do sistema de TA a ser re-treinado, os autores relataram uma melhoria nos valores de BLEU tanto para a nova tradução do mesmo texto (BLEU passou de 23,50 para 25,73) quanto para a tradução de novos textos (BLEU passou de 25,27 para 25,51).

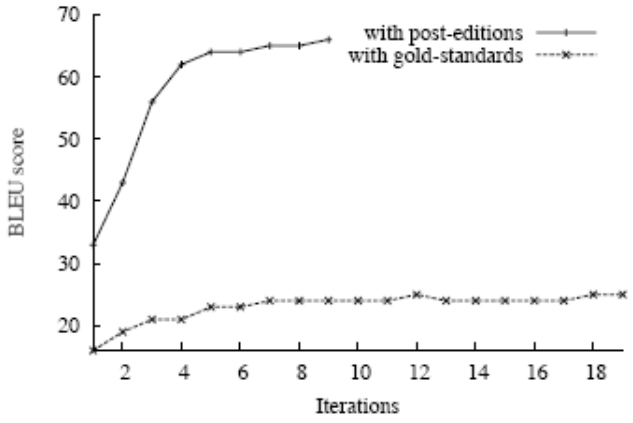
- **Correção automática da saída do sistema de TA (pós-edição)**: um modelo de tradução foi treinado recebendo como entrada a saída da TA e suas pós-edições efetuadas por humanos. Ao passar a saída da TA no novo modelo de tradução (sistema de pós-edição), a nova tradução dos mesmos textos teve um ganho em BLEU (de 23,50 para 24,58), enquanto a tradução de novos textos demonstrou uma queda nos valores de BLEU (de 25,27 para 24,32).

- **Reajuste dos pesos do modelo *log-linear***: nessa estratégia, ao invés de usar as traduções de referência no corpus de otimização (*tuning*), foram utilizadas as traduções pós-editadas. Essa decisão está baseada no fato de que as traduções de referência são, muitas vezes, mais distantes da saída original da TA do que as traduções pós-editadas. Com isso, após 9 iterações, o BLEU passou de 33 para 66 usando a saída pós-editada e precisou de 19 iterações para passar de 23 a 25 usando a referência. A Figura 2.7 mostra a evolução do BLEU ao se usar a otimização de pesos do modelo com MERT, usando a referência tradicional (*gold-standards*) e textos pós-editados como referência (*post-editions*). Usando o corpus de teste observou-se que não houve uma diferença significativa nos valores de BLEU entre as duas estratégias de otimização. Sem a otimização o BLEU foi de 25,27. Usando a otimização feita com textos pós-editados o BLEU aumentou para 25,36 e na otimização usando a referência tradicional o BLEU aumentou um pouco mais, para 25,43.

A correção automática baseada em modelos estatísticos pode ser empregada também para melhorar a saída de textos processados por modelos de tradução baseados em regras (RBMT). Em (LAGARDA et al., 2009) e (SIMARD; GOUTTE; ISABELLE, 2007) a utilização de sistemas RBMT + APE melhorou significativamente o desempenho do modelo de RBMT. Ambos os experimentos foram avaliados utilizando métricas conhecidas na TA como TER e BLEU, além de outros critérios de caráter mais subjetivo. Os resultados relatados por esses trabalhos são apresentados nas Tabelas 2.9, 2.10 e 2.11.

No primeiro experimento apresentado por Lagarda et al. (2009), foram escolhidos dois corpora: o Parliament, que contém transcrições de discursos parlamentares; e o Protocols, que contém protocolos médicos. Tanto na avaliação automática das traduções com base em TER e BLEU (apresentada na Tabela 2.9) quanto na avaliação humana (apresentada na Tabela 2.10) pôde-se concluir que o uso da APE em sistemas de TA baseados em RBMT melhora signifi-

Figura 2.7: Mudanças nos valores de BLEU durante a otimização de pesos com MERT usando sentenças pós-editadas (*post-editions*) ou de referência (*gold-standards)* no APE desenvolvido por Potet et al. (2011).



Fonte: (POTET et al., 2011, p. 166).

cativamente a qualidade da tradução gerada quando comparada àquela com o uso de RBMT somente. Aqui, vale lembrar que quanto maior o valor de BLEU e menor o valor de TER, melhor.

Tabela 2.9: Avaliação automática nos corpora Parliament e Protocols do APE desenvolvido por Lagarda et al. (2009).

	Parliament		*Protocols*	
	BLEU	TER	BLEU	TER
RBMT	29,1	46,7	29,5	48,0
RBMT+APE	48,4	35,9	33,6	46,2

Fonte: (LAGARDA et al., 2009).

Os valores apresentados na Tabela 2.10 têm como base uma medida binária proposta por Lagarda et al. (2009) que se baseia no trabalho de avaliação humana de fatores qualitativos de (CALLISON-BURCH et al., 2007). A medida binária em questão, a aceitabilidade (*suitability*), permite que os avaliadores humanos classifiquem a tradução como aceitável (*suitable*) ou não (*not-suitable*). Uma tradução aceitável é aquela que, de acordo com a avaliação do tradutor humano, pode ser pós-editada de forma a aumentar sua qualidade. Por outro lado, se o tradutor humano preferir ignorar a tradução proposta para a sentença e começar novamente, esta é avaliada como não-aceitável (*not-suitable*).

A partir dos resultados apresentados na Tabela 2.10 é possível notar como um APE é útil para melhorar a qualidade de uma tradução automática RBMT ruim, tornando-a aceitável. Algo semelhante é verificado no segundo estudo de Simard, Goutte e Isabelle (2007), descrito na Tabela 2.11, no qual conclui-se que um APE baseado em SMT é uma excelente alternativa para melhorar a saída de um sistema de RBMT genérico, além de ser bastante viável à custosa

Tabela 2.10: Avaliação humana para os corpora Parliament e Protocols. Porcentagem das sentenças consideradas aceitáveis (*suitable*) para o *baseline* (RBMT) e o APE desenvolvido por Lagarda et al. (2009).

	Parliament	*Protocols*
RBMT	68%	60%
RBMT+APE	94%	67%

Fonte: (LAGARDA et al., 2009).

pós-edição humana.

Tabela 2.11: Avaliação automática no corpus Job Bank do *baseline* (RBMT) e o APE desenvolvido por Simard, Goutte e Isabelle (2007).

	inglês-francês		francês-inglês	
	BLEU	TER	BLEU	TER
RBMT	32,9	53,5	31,2	59,3
RBMT+APE	41,6	47,3	44,9	41,0

Fonte: Adaptado de (SIMARD; GOUTTE; ISABELLE, 2007).

O uso de um sistema SMT monolíngue para traduzir de um inglês ruim para um inglês bom também foi aplicado por (SENEFF; WANG; LEE, 2006) em um sistema de TA por interlíngua. Para gerar o corpus de treinamento do sistema SMT a ser usado como APE, o corpus disponível foi traduzido de inglês para chinês e de volta para inglês. Apesar dos experimentos conduzidos em (SENEFF; WANG; LEE, 2006) indicarem que o número de traduções boas subiu com essa abordagem, ela gerou também um aumento nas traduções ruins. A Tabela 2.12 mostra as avaliações feitas por humanos para um conjunto de 409 sentenças traduzidas de chinês para inglês. Nessa avaliação, as notas vão de 5 (uma tradução perfeita), passando pela 3 (aceitável), e chegando a 1 (incorreta tanto semântica como sintaticamente).

Tabela 2.12: Avaliação humana em 409 sentenças traduzidas do chinês para o inglês por um sistema SMT, o *baseline* (Interlíngua) e o APE desenvolvido por Seneff, Wang e Lee (2006).

	5	**4**	**3**	**2**	**1**	**Média**
SMT	265	5	38	6	95	3,83
Interlíngua	270	28	83	8	20	4,27
Interlíngua+APE	333	10	33	7	26	4,51

Fonte: Adaptado de (SENEFF; WANG; LEE, 2006).

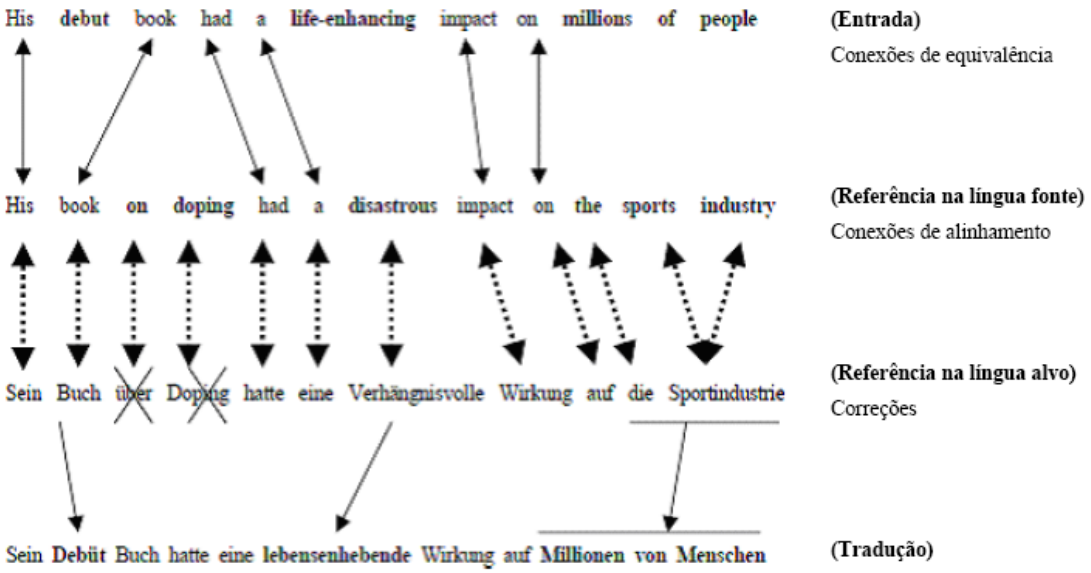
A partir do que foi exposto nesta seção, pode-se concluir que as estratégias que usam SMT para pós-editar a saída da TA mostraram-se efetivas quando aplicadas à saída da tradução tanto de sistemas SMT quanto de sistemas RBMT.

2.3.2.2 Correção automática auxiliada por memórias de tradução

Esse tipo de APE possibilita que o usuário de sistemas de TA ou técnicas de AM alimentem a memória de tradução (*Translation Memory*, TM) com regras que poderão ser reutilizadas em traduções futuras. Uma memória de tradução contém partes de textos em seu formato original, ou seja, na língua fonte, e seu correspondente traduzido na língua alvo. Quando deseja-se produzir uma nova tradução, a sentença de entrada é comparada com a sentença correspondente na língua fonte (armazenada na TM juntamente com seu par na língua alvo) a fim de encontrar a tradução.

O uso da TM em um APE pode ser diversificado. Em (KRANIAS; SAMIOTOU, 2004), os autores propõem métodos para inserção, exclusão e substituição de palavras com o objetivo de encontrar a sentença correspondente exata na língua alvo, ainda que nenhum dos pares de sentenças armazenados equivalham exatamente àquela a ser traduzida. O método, classificado pelos autores como EBMT faz o "casamento" dos segmentos a serem traduzidos caso encontre uma tradução idêntica ou similar (chamado pelos autores de *fuzzy match*) na memória de tradução. Quando nenhum casamento é encontrado, então o tradutor automático é chamado. A ilustração de seu funcionamento pode ser vista na Figura 2.8, tendo como exemplo a tradução de uma sentença do idioma inglês para o alemão. Apesar de não ser um APE propriamente dito, pois é utilizado no pré-processamento, a técnica poderia ser aplicada em um APE.

Figura 2.8: Exemplo do uso da TM.



Fonte: Adaptado de (KRANIAS; SAMIOTOU, 2004, p. 332).

Exemplos de implementação e estudos relacionados a processamento utilizando TM para o português do Brasil podem ser encontrados em (GOMES; PARDO, 2008) ou no Portal de Tradução

Automática, PorTAl, que dispõe de uma ferramenta de memória de tradução para uso *on-line* (KAWAMORITA; CASELI, 2012)[25].

A fim de medir o quanto a ferramenta auxilia um tradutor humano na pós-edição de textos traduzidos previamente por sistemas de TA, em comparação à pós-edição dos mesmos textos sem o uso de ferramenta alguma, em (GOMES; PARDO, 2008) foi pedido aos usuários para que corrigissem um texto o mínimo possível até que ficasse coerente e gramaticalmente correto. Os usuários do teste possuíam bom domínio nos idiomas português brasileiro e inglês (os testes foram realizados com traduções de textos jornalísticos do inglês para o português brasileiro) e além deles, um avaliador experiente também corrigiu manualmente os textos para que sua correção fosse tomada como base. As medidas obtidas, e demonstradas na Tabela 2.13, comparam o número de correções feitas pelos usuários com as feitas pelo avaliador experiente (referência). A precisão é calculada como o número de erros corrigidos pelo usuário dividido pelo número de erros corrigidos da mesma maneira na referência. Para a cobertura, toma-se o número de erros corrigidos pelo usuário e divide-se pelo número de erros cuja correção é esperada (presentes na referência). Finalmente tem-se a medida-F que é calculada como (2*Precisão*Cobertura) / (Precisão + Cobertura).

Tabela 2.13: Avaliação quantitativa das alterações realizadas com o auxílio da ferramenta de TM desenvolvida por Gomes e Pardo (2008) em um texto de 21 sentenças.

Tradutor Automático	Precisão %	Cobertura %	Medida-F
Babelfish[26]	81	86	84
Google	89	92	90

Fonte: (GOMES; PARDO, 2008).

Ainda em (GOMES; PARDO, 2008), após terem sido feitas as pós-edições no primeiro texto, a ferramenta recebeu um novo texto de uma outra agência de notícias, a respeito da mesma notícia. O novo texto foi traduzido pelo mesmo sistema de TA que havia sido utilizado no texto anterior. Nessa etapa a ferramenta sugeriu 19 modificações automáticas, especialmente para palavras não traduzidas no texto original e que apareceram novamente e também para erros gramaticais comumente gerados pelo tradutor automático escolhido. Os usuários, através de um questionário de usabilidade, avaliaram de maneira positiva a ferramenta e mencionaram que ela facilitou o trabalho de revisão de textos traduzidos automaticamente.

2.3.2.3 Correção automática com o uso de verificadores gramaticais

Entre os trabalhos que seguem esta abordagem, os três citados a seguir se mostraram mais relevantes para esta pesquisa. Em (DOYON et al., 2008) são demonstrados resultados de experi-

[25] Disponível em: <http://www.lalic.dc.ufscar.br/portal>. Acesso em: 21 jan. 2014.
[26] Atualmente disponível em: <http://br.bing.com/translator/>. Acesso em: 21 jan. 2014.

mentos com três diferentes tradutores do tipo RBMT utilizando pós-edição humana e também pós-edição automática realizada com diversos pós-editores disponíveis comercialmente.

Em um desses estudos é comparada a aceitabilidade da saída de três tradutores automáticos diferentes e sua saída pós-editada com o editor *WordPerfect*[27], ajustado com as configurações ótimas citadas no artigo em questão, onde somente as modificações automáticas do seu verificador gramatical, que se mostraram benéficas em testes anteriores, foram habilitadas. Além disso, são efetuadas comparações com pós-edições humanas da saída da TA realizadas com o método de edição completa e edição breve. No método de edição completa, editores profissionais, tendo o inglês como língua nativa, receberam somente a saída da TA (sem ter acesso ao texto fonte) com o intuito de transformá-las em textos publicáveis. No método de edição breve, editores de uma empresa de tradução, receberam a saída da TA e o texto fonte que poderia ser usado como referência, além disso usaram também as diretrizes da companhia para a execução de pós-edição comercial que prioriza a precisão ao invés da perfeição, ou seja, realizaram apenas as correções necessárias para permitir o entendimento do texto.

A tarefa de avaliação humana foi conduzida tanto por analistas (engenheiros, assistentes administrativos, gerentes e outros profissionais que não eram tradutores profissionais) como por tradutores profissionais. Os analistas que participaram do experimento avaliaram gramaticalmente a saída da TA em traduções do árabe para o inglês. Estavam disponíveis para a avaliação: a tradução sem pós-edição e a versão pós-editada. A tarefa dos analistas era atribuir notas de 1 a 7 sobre a aceitabilidade da gramática em determinadas passagens selecionadas aleatoriamente. A eles também foi pedido que opinassem sobre qual nível da escala de 1 a 7 indicava um documento útil. Já aos tradutores profissionais cabia indicar se preferiam: editar a saída da TA, utilizá-la como base para criar sua própria tradução ou realizar a tradução sem utilizar a saída da TA apresentada.

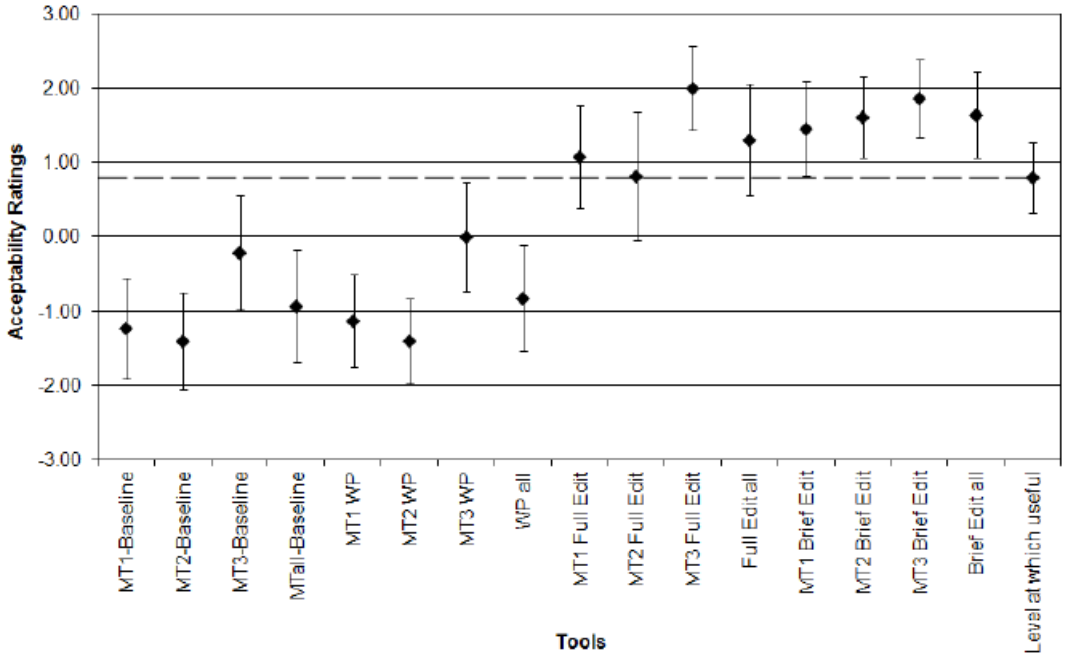
As notas de aceitabilidade dadas pelos analistas e o nível de aceitabilidade foram codificados entre +3 (extremamente aceitável) e -3 (extremamente inaceitável). A Figura 2.9 mostra o resultado contendo a média e o desvio padrão. Como é possível notar, a pós-edição usando verificador gramatical (**WP) melhorou levemente o *baseline* (**Baseline), mas ainda ficou bem abaixo da aceitabilidade relatada para as pós-edições realizadas por humanos (**FullEdit e **BriefEdit).

A Figura 2.10 mostra a porcentagem de aceitabilidade por parte dos tradutores humanos, que leva em conta tanto as respostas de que usariam o texto traduzido para edição ou como base para realizar suas traduções.

Ao final dos experimentos realizados em (DOYON et al., 2008) concluiu-se que, apesar do

[27] Versão de avaliação disponível em: <http://www.corel.com/corel/product/index.jsp?pid=prod4720105>. Acesso em: 21 jan. 2014.

Figura 2.9: Aceitação, por parte de analistas humanos, de traduções feitas por três diferentes tradutores automáticos (MT1, MT2, MT3) e a média entre eles (MTAll): sem pós-edição (*-Baseline*), pós-editadas utilizando *WordPerfect* (WP), pós-editadas por humanos com os métodos de edição completa (*Full Edit*) e edição breve (*Brief Edit*).



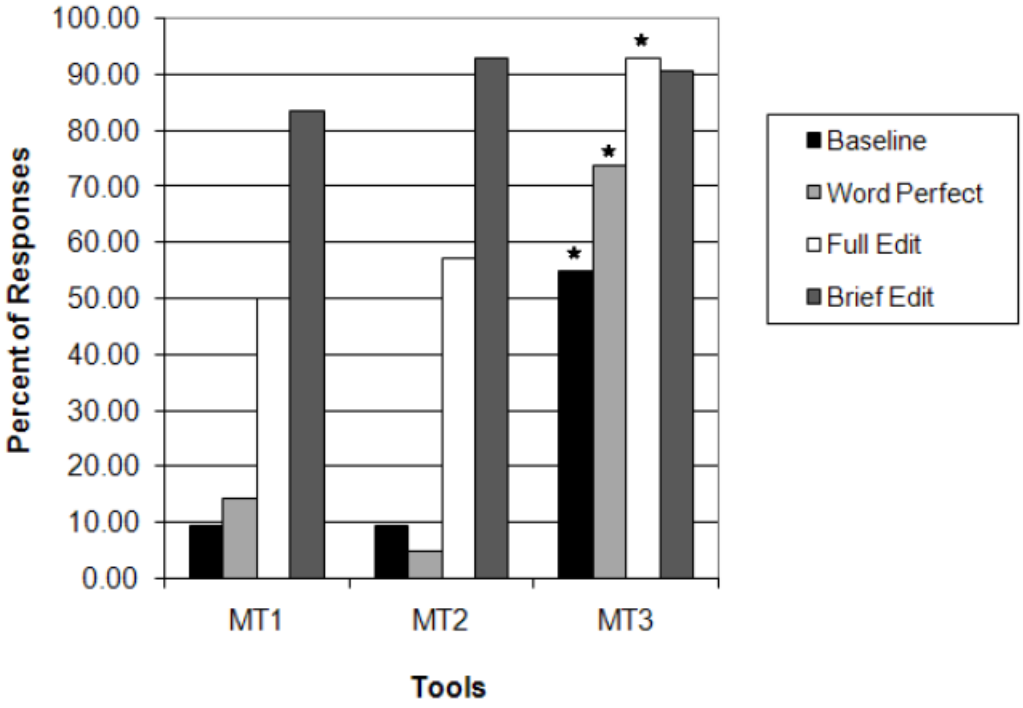
Fonte: (DOYON et al., 2008, p. 350).

uso do pós-editor comercial *WordPerfect* ter gerado um impacto positivo na aceitação da saída da TA por parte dos analistas e dos tradutores profissionais participantes do estudo, a pós-edição feita por humanos ainda é a única que eles escolheriam utilizar.

Em (STYMNE, 2011b) é apresentada uma abordagem que envolve o pré-processamento e depois o pós-processamento de textos traduzidos automaticamente para línguas germânicas. No pós-processamento, que é o objeto de estudo desta pesquisa, são usadas sugestões de um verificador gramatical (tanto o verificador e a forma como foi usado estão detalhados em (STYMNE; AHRENBERG, 2010)) para encontrar e corrigir erros na saída de um tradutor estatístico. A pós-edição, neste caso, melhorou alguns tipos de erros, por exemplo, os relacionados à falta de concordância e à ordem incorreta das palavras.

As melhoras absolutas em BLEU e TER reportadas em (STYMNE; AHRENBERG, 2010) após o pós-processamento foram de no máximo 0,18 e 0,09, respectivamente. As baixas taxas de melhoria, tanto em BLEU como em TER, podem ser explicadas pelo fato de que apenas uma pequena proporção das sentenças sofreu pós-edição. Analisando-se somente as sentenças que sofreram pós-edição, 68-74% das correções que foram feitas foram consideradas úteis e apenas 10% pioraram a qualidade da sentença. A fim de conseguir uma abrangência maior na verificação de erros pelo analisador gramatical usado e consequentemente pós-editar mais

Figura 2.10: Aceitação, por parte de tradutores profissionais, de traduções feitas por três diferentes tradutores automáticos (MT1, MT2, MT3), nesta ordem: sem pós-edição (*Baseline*), pós-editadas utilizando *WordPerfect*, pós-editadas por humanos com os métodos de edição completa (*Full Edit*) e edição breve (*Brief Edit*).



Fonte: (DOYON et al., 2008, p. 352).

sentenças, uma das sugestões de Stymne e Ahrenberg (2010) é desenvolver um novo checador gramatical voltado a erros de TA, pois o que foi utilizado havia sido originalmente desenvolvido para encontrar erros cometidos por humanos.

Para o português do Brasil, o verificador gramatical do MS-Word 2010 foi aplicado em (AVANÇO; NUNES, 2013) para a correção de erros de TA. A configuração padrão do verificador foi mantida e os erros que tivessem uma sugestão de correção apresentada por ele foram tratados a fim de que a correção fosse feita de forma automática. Os testes realizados trataram da correção de erros de concordância nominal e de concordância verbal de traduções feitas pelo TAEIP, a partir do idioma inglês. Quando as correções automáticas foram aplicadas houve um aumento de 1,14% em BLEU e 0,45% em NIST na avaliação de 35 sentenças para concordância nominal; e de 1,12% em BLEU e 0,37% em NIST na avaliação de 40 sentenças para concordância verbal.

2.3.2.4 Correção automática com o uso de algoritmos de AM

Uma técnica de AM aplicada na correção de erros de TA é a TBL usada em (ELMING, 2006) para pós-editar a saída de um sistema RBMT. Em tal trabalho foram usados como entrada para

o TBL 34 textos totalizando 265.000 palavras aproximadamente, divididas aleatoriamente em três subconjuntos mostrados na Tabela 2.14[28].

Tabela 2.14: Subconjuntos resultantes da divisão do corpus usado em (ELMING, 2006).

Corpus	Textos	Sentenças	Palavras
Treinamento	26	12000	220000
Validação	4	2000	25000
Teste	4	1200	20000

O papel do TBL, nesse experimento, é aprender automaticamente regras de correção de erros usando inicialmente uma lista já contendo algumas regras instanciáveis (*templates*), pois o TBL não cria regras de transformação sem que haja uma base para isso. As regras, contidas em 70 *templates*, são sobre a possível influência contextual na substituição – em qual contexto uma palavra é trocada por outra – usando as 6 palavras mais próximas, 3 palavras de cada lado.

Além disso, para que sejam instanciadas como regras concretas, cada *template* candidato deve seguir duas outras restrições: fazer correções corretas em pelo menos 50% dos casos, e também gerar pelo menos três correções corretas a mais do que incorretas. Se nenhuma das candidatas a regras obedecer essas duas últimas restrições, o algoritmo termina o aprendizado. As regras aprendidas são baseadas em etiquetas morfossintáticas (*part-of-speech tags*) e formas superficiais (palavras ou *tokens* da maneira como ocorrem no texto). Ao aplicar as regras aprendidas houve um aumento em BLEU, em relação ao desempenho do sistema RBMT sendo avaliado, de 72,2 para 73,6 usando o corpus de validação, 59,5 para 63,5 usando o conjunto de teste e de 64,6 para 67,6 usando as sentenças do corpus de validação e as sentenças do corpus de teste.

Ainda aplicada à saída de um sistema RBMT, George e Japkowicz (2005) buscam e tratam os erros de pronomes relativos em traduções do francês para o inglês usando para isso três diferentes técnicas de AM: Naive Bayes, Árvore de Decisão (ambas descritas em mais detalhes na seção 2.2) e One Rule (HOLTE, 1993). Um corpus contendo pronomes relativos incorretamente traduzidos e suas correções correspondentes é usado para treinar o sistema proposto que foi dividido nas duas etapas comuns a esta pesquisa: identificação de erros e correção de erros. Na primeira etapa, busca-se identificar se uma sentença é boa ou ruim, passando somente as ruins para a segunda etapa, na qual são aplicadas características sintáticas e semânticas com o propósito de corrigir automaticamente os pronomes. A experiência mostrou que 83,72% das sentenças foram propriamente identificadas como incorretas e 73,07% delas corrigidas corretamente. Assim como a pesquisa aqui apresentada, George e Japkowicz (2005) realizaram a pós-edição em duas etapas (identificação e correção), contudo apenas para erros de pronomes

[28] Os valores mostrados na tabela 2.14 são aproximados.

relativos, categoria não especificamente investigada nesta pesquisa.

Tomando como base classificações de erros e pós-edições previamente realizadas por humanos, Llitjós (2007) automaticamente amplia e refina a gramática e o léxico de um sistema de TA por transferência (RBMT), conseguindo obter melhorias nesse sistema. O sistema de refinamento das regras, chamado de ARR (*Automatic Rule Refinement*) é composto do módulo refinador de regras em si e da ferramenta usada na classificação e correção da tradução por humanos, chamada de TCTool, onde o usuário faz a correção ou o alinhamento palavra a palavra. O módulo refinador de regras usa a saída da TCTool, com os erros classificados e corrigidos por humanos, para produzir árvores sintáticas. Em seguida, ele as compara com as árvores de tradução produzidas pelo sistema de TA, recuperando assim regras relevantes que devem ser refinadas.

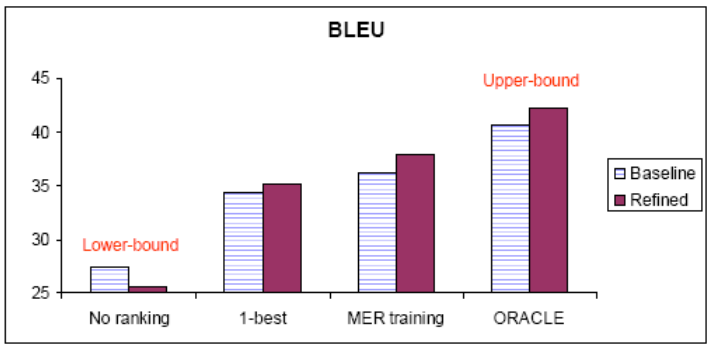
O sistema de TA sendo refinado possui um mecanismo de transferência, um verificador de fragmentação (*fragmentation penalty*) e um decodificador. O mecanismo de transferência combina a gramática e o léxico para produzir todas as possíveis traduções da língua fonte para a língua alvo. Já o papel do verificador de fragmentação é gerar um *ranking* das traduções alternativas com base em quantas vezes a sentença traduzida tenha que passar por *parsing* parcial para que seja totalmente processada – a ideia é que quanto mais *parsing* for preciso, pior é a sentença sendo processada. Finalmente o decodificador, que possui um modelo de língua estatístico, é usado para selecionar a melhor tradução alternativa de uma lista final de traduções candidatas geradas pelo mecanismo de transferência.

Os experimentos realizados com o ARR produziram resultados diferentes de BLEU. Um experimento *No Ranking* foi feito sem o decodificador e a primeira sentença gerada pelo mecanismo de transferência era a escolhida com o objetivo de definir um patamar inferior (*lower-bound*) para fins de comparação com os outros. Para definir um patamar superior (*upper-bound*) foi realizado o experimento ORACLE no qual, a partir de uma lista das 100 melhores sentenças, a melhor, segundo o BLEU, foi selecionada. A variação *1-best* combinou o mecanismo de transferência, o verificador de fragmentação e o decodificador. Uma otimização[29] do decodificador usando MERT configurado para melhorar o BLEU foi executada para o experimento MER *Training*. Os resultados de BLEU nos experimentos com inglês-espanhol apresentados pela ferramenta podem ser vistos na Figura 2.11.

Pode-se observar, na Figura 2.11, que nos experimentos, exceto no *No Ranking*, o sistema de TA com refinamento teve melhora em BLEU quando comparado ao *baseline*. Apesar de não ser um sistema de APE propriamente dito, pois é utilizado para refinar regras de um sistema

[29] A otimização dos pesos do decodificador é feita seguindo um processo de *hill-climbing*, gerando um novo *ranking* na lista das *n* melhores, o que melhorará a equiparação entre a primeira melhor tradução e as traduções de referência dadas (LLITJÓS, 2007).

Figura 2.11: Resultados de BLEU para as diferentes variações do *baseline* sem e com o *Automatic Rule Refinement* no APE desenvolvido por Llitjós (2007).



Fonte: Adaptado de (LLITJÓS, 2007, p. 144).

de TA, a técnica pode ser adaptada a fim de ser aplicada em um APE com AM. Llitjós menciona como um dos trabalhos futuros a possibilidade de investigar uma abordagem que processe dados extraídos da ferramenta TCtool (parte do ARR) e, usando técnicas de AM, aprenda uma gramática que gere traduções corretas a partir de traduções incorretas.

2.3.2.5 Resumo de trabalhos de correção automática de erros

As tabelas 2.15, 2.16, 2.17 e 2.18 trazem um resumo das principais características dos trabalhos de correção automática de erros citados anteriormente.[30]

2.4 Considerações finais

Tanto na identificação de erros (ou avaliação) automática, envolvendo escalas ou categorização de erros, como na correção automática, ainda há muito a evoluir até que ferramentas criadas com esses propósitos sejam aceitas por tradutores profissionais que trabalham com pós-edição da TA.

Alguns trabalhos pesquisados demonstram resultados animadores, mas em outros casos os resultados foram tímidos ou até mesmo piores com relação aos sistemas usados como base de comparação (*baseline*). Trabalhar para fazer com que as ferramentas existentes ou novas consigam competir com a anotação de erros feita por humanos e sua correção, torna a identificação e a correção automáticas de erros de TA um desafio muito interessante, pois é um campo cheio de possibilidades para melhoria.

[30] Os idiomas citados nas tabelas são ar: árabe, ch: chinês, , da: dinamarquês, de:alemão, en: inglês, es:espanhol, fr:francês, se: sueco

[31] Disponível em: <http://www.jobbank.gc.ca>. Acesso em: 21 jan. 2014.

[32] Disponível em: <http://www.esteam.se/solutions/translation-automation.html>. Acesso em: 21 jan. 2014.

Tabela 2.15: Resumo dos trabalhos referentes à correção automática baseada em SMT.

Referência	Idiomas	Saída da TA	Corpus	Abordagem
(BÉCHARA; MA; GENABITH, 2011)	en-fr, fr-en	SMT	Memória de tradução de uma empresa de informática (Symantec)	SMT treinado usando a própria saída da TA e sua correspondente referência para criar um tradutor monolíngue
(UENISHI, 2013)	en-pt	SMT	FAPESP-v2 para treinamento e teste (teste-a e teste-b) (AZIZ; SPECIA, 2011)	SMT treinado usando a própria saída da TA e sua correspondente referência para criar um tradutor monolíngue
(POTET et al., 2011)	fr-en	SMT	*Europarl* + *News Commentary* para o tradutor e 175 segmentos pós-editados para o pós-editor automático	SMT treinado novamente com saída da TA + saída da TA pós-editada (e outros dois métodos)
(LAGARDA et al., 2009)	en-es	RBMT	*Parliament* (transcrições de discursos parlamentares) e *Protocols*(conjunto de protocolos médicos)	SMT treinado com saída do sistema de RBMT
(SIMARD; GOUTTE; ISABELLE, 2007)	en-fr, fr-en	RBMT	*Job Bank*: corpus contendo anúncios de empregos com traduções automáticas pós-editadas. Ali constam o texto enviado pelo empregador, a saída da tradução automática e a saída pós-editada, que é a que aparece na página no idioma alternativo, ou inglês ou francês (idiomas oficiais canadenses.[31]	SMT treinado com corpus paralelo contendo a saída original do sistema RBMT + textos pós-editados
(SENEFF; WANG; LEE, 2006)	ch-en	TA por interlíngua	Corpus no domínio da aviação	SMT treinado com corpus paralelo contendo traduções de inglês ruim para inglês bom

Tabela 2.16: Resumo dos trabalhos referentes à correção automática auxiliada por TM.

Referência	Idiomas	Saída da TA	Corpus	Abordagem
(KRANIAS; SAMIOTOU, 2004)	en-de, en-fr	O sistema foi implementado na ferramenta *ESTeam Translator (ET) Language Toolbox*[32]	20.000 sentenças em inglês a serem traduzidas para alemão e francês	O método, classificado pelos autores como EBMT, usa TM. Faz o "casamento" dos segmentos a serem traduzidos caso encontre uma tradução idêntica ou similar (chamado pelo autor de *fuzzy match*). Quando nenhum casamento é encontrado, então o tradutor automático é chamado
(GOMES; PARDO, 2008)	en-pt(br)	SMT	Textos jornalísticos	TM contendo fragmento fonte, tradução automática e fragmento editado alinhados entre si

Dentre os trabalhos citados neste capítulo, os seguintes serviram de base para a pesquisa aqui apresentada: (POPOVIC; BURCHARDT, 2011), no qual foram baseadas as categorias de erros investigadas nesta pesquisa; (FELICE; SPECIA, 2012), do qual foram retiradas algumas ideias

Tabela 2.17: Resumo dos trabalhos referentes à correção automática usando verificadores gramaticais

Referência	Idiomas	Saída da TA	Corpus	Abordagem
(DOYON et al., 2008)	ar-en	TA por transferência	Traduções de árabe para inglês de notícias transmitidas via rádio ou TV	Pós-editor comercial do *WordPerfect* (configurado segundo informado em (DOYON et al., 2008))
(STYMNE, 2011b)	en-de, en-se, en-da	Textos pré-processados e passados por SMT	*Europarl* para alemão e sueco e *Automotive* para dinamarquês	Usa sugestões de um verificador gramatical (verificador e uso estão detalhados em (STYMNE; AHRENBERG, 2010)) para encontrar erros e corrigir erros
(AVANÇO; NUNES, 2013)	en-pt	SMT	FAPESP-v2 (teste-a) (AZIZ; SPECIA, 2011)	Aplica o verificador gramatical do MS-Word 2010 para obter sugestões de correção e a partir delas realizar pós-edições

Tabela 2.18: Resumo dos trabalhos referentes à correção automática usando AM.

Referência	Idiomas	Saída da TA	Corpus	Abordagem
(ELMING, 2006)	en-da	RBMT	corpus paralelo contento textos traduzidos automaticamente no domínio de patentes químicas e a versão traduzida corrigida por humanos	TBL
(GEORGE; JAPKOWICZ, 2005)	fr-en	RBMT	Conjunto de dados com 72 sentenças contendo, pelo menos, um erro de pronome relativo e possivelmente outros tipos de erro	Naive Bayes, árvore de decisão e 1R
(LLITJÓS, 2007)	en-es	TA por transferência	*Diagnostic set* (DSet) para desenvolvimento, *AVENUE Elicitation Corpus* (EC) para validação e *Basic Travel Expression Corpus* (BTEC) para teste	Usa saída de TM, corrigida por humanos, para produzir árvores sintáticas, compara-as com as árvores de tradução produzidas pelo sistema de TA, recuperando regras relevantes que devem ser refinadas

para as *features* de AM usadas na identificação de erros; e (ELMING, 2006), do qual derivou-se a técnica de AM usada na geração de regras de correção de erros.

Capítulo 3

ANOTAÇÃO DO CORPUS DE TREINAMENTO

Esse capítulo descreve o processo de anotação manual do corpus (MARTINS et al., 2013) utilizado no treinamento dos algoritmos de AM aplicados para a realização das duas etapas do processo de pós-edição automática da tradução: identificação de erros e correção de erros. Para tanto, as próximas seções descrevem: o corpus de treinamento (3.1), as categorias de erros (3.2) e as regras (3.3) definidas para a anotação, bem como os resultados desse processo (3.4).

3.1 Corpus de treinamento

Como o APE projetado nesta pesquisa visa a correção de erros produzidos na tradução de textos em inglês para o português do Brasil, o corpus de treinamento dos algoritmos de AM contém os textos da revista Pesquisa FAPESP[1] (AZIZ; SPECIA, 2011)[2] correspondentes ao conjunto teste-a contendo 1.314 pares de sentenças paralelas. As sentenças em inglês (Fonte ou Src) deste corpus estão acompanhadas de sentenças em português geradas por humanos (Referência ou Ref). As sentenças em inglês foram traduzidas usando o tradutor automático TAEIP treinado previamente para ser incorporado ao PorTAl conforme descrito na seção 5.5. As sentenças traduzidas pelo TAEIP (Tradução ou Sys) foram incorporadas às demais formando os trios de sentenças do corpus de treinamento, como ilustra a Figura 3.1.

O corpus de treinamento foi processado com os etiquetadores morfossintáticos de Apertium[3] (ARMENTANO-OLLER et al., 2006) enriquecidos com dados linguísticos para o português do Brasil e o inglês no projeto ReTraTos (CASELI, 2007)[4], bem como o alinhador de palavras GIZA++[5] (OCH; NEY, 2003). A Figura 3.2 mostra as mesmas sentenças da Figura 3.1 alinha-

[1]Disponível em: <http://revistapesquisa.fapesp.br/>. Acesso em: 19 fev. 2014.

[2]Disponível em: <http://pers-www.wlv.ac.uk/∼in1676/resources/fapesp/index.html>. Acesso em: 19 fev. 2014.

[3]Disponível em: <http://www.apertium.org/> . Acesso em: 20 jan. 2014.

[4]Disponível em: <http://www.lalic.dc.ufscar.br/portal>. Acesso em: 19 fev. 2014.

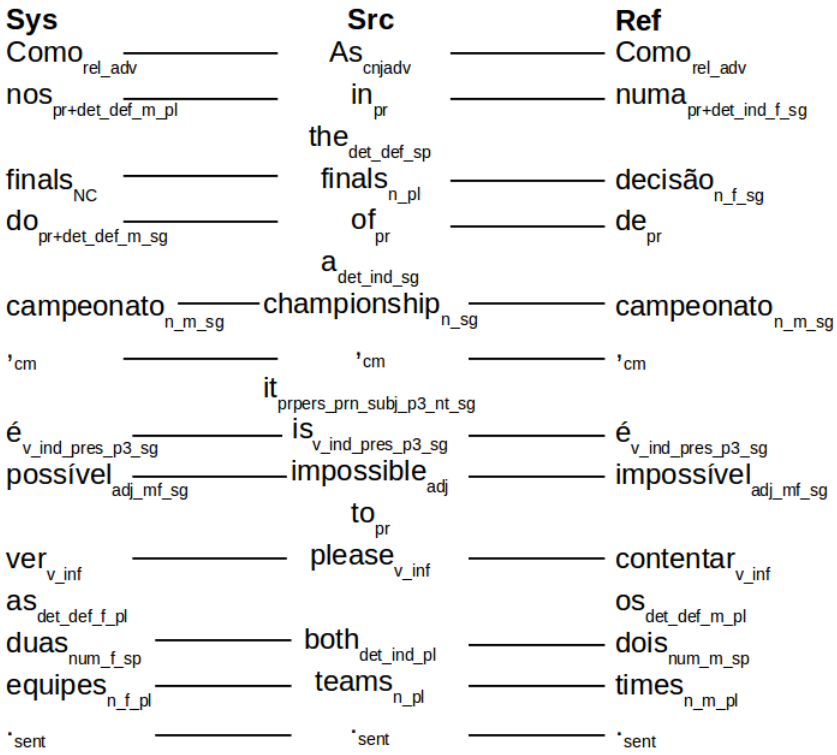
[5]Disponível em: <http://www.statmt.org/moses/giza/GIZA++.html>. Acesso em: 20 jan. 2014.

Figura 3.1: Exemplo de um trio de sentenças do corpus de treinamento: sentença fonte em inglês (Src), sentença traduzida para o português (Sys) e tradução de referência em portugueês (Ref). Os erros de TA anotados estão sublinhados em Src, Sys e Ref.

Src	As in the finals of a championship , it is impossible to please both teams .
Sys	Como nos finals do campeonato , é possível ver as duas equipes .
Ref	Como numa decisão de campeonato , é impossível contentar os dois times .

das e enriquecidas com dados morfossintáticos. O alinhamento e os dados morfossintáticos são utilizados para a geração das *features* a serem usadas nos experimentos com identificação e correção de erros apresentados nos capítulos 4 e 5, respectivamente.

Figura 3.2: O mesmo trio de sentenças paralelas da Figura 3.1 com informação morfossintática e alinhamento. NC indica um valor não conhecido.



3.2 Categorias de erro

As categorias de erro selecionadas para a anotação do corpus de treinamento foram baseadas em (POPOVIC; BURCHARDT, 2011) que, por sua vez, baseiam-se em algumas categorias de (VILAR et al., 2006). Nos exemplos apresentados a seguir, Fonte representa a sentença original fornecida como entrada para o tradutor automático, Referência é a sentença na língua alvo gerada por um humano como a tradução de Fonte e Tradução é a saída do tradutor automático quando Fonte foi fornecida como entrada.

A. Erros sintáticos (*inflectional errors* de Popovic e Burchardt (2011))

- Escopo de marcação: engloba apenas uma palavra.
- Descrição: a palavra na qual o erro ocorre tem a forma base (lema) correta (comparando-se a Tradução com a Referência), mas a forma superficial está errada.
- Forma de anotação: marcar a palavra incorreta em Fonte, Tradução e Referência. Na dúvida, não marcar em Fonte ou Referência.

A.1 – Concordância em número: erros de concordância de singular e plural. Exemplo:
Fonte: The girls went to school.
Referência: As garotas foram à escola.
Tradução: A garotas foi à escola.

A.2 – Concordância em genêro: erros de concordância para masculino e feminino.
Exemplo:
Fonte: The girl went to school.
Referência: A garota foi à escola.
Tradução: O garota foi à escola.

A.3 – Flexão verbal: verbos que apresentam conjugação ou forma verbal incorreta.
Exemplo:
Fonte: The girls went to school
Referência: As garotas foram à escola
Tradução: As garotas iam à escola

A.4 – PoS (*Part of Speech*): mudança de categoria, por exemplo, "sonhar" traduzido como "sonho". Aqui houve mudança de verbo para substantivo.

B. Erros lexicais (erros de natureza lexical)

- Escopo de marcação: engloba apenas uma palavra.
- Descrição: a palavra na qual o erro ocorre não compartilha a forma base (lema) com nenhuma palavra na Referência.
- Forma de anotação: marcar a palavra incorreta em Fonte, Tradução e Referência. Na dúvida, não marcar em Fonte ou Referência.

B.1 – Palavra extra (*extra words* de Popovic e Burchardt (2011)): uma palavra na Tradução que não tem nenhuma correspondência em Fonte.
Fonte: The girl went to school
Referência: A garota foi à escola
Tradução: A garota foi a sua escola

Neste caso é importante sempre olhar para a sentença fonte para determinar se houve inserção de alguma palavra como no exemplo a seguir no qual a Tradução está correta.

Fonte: The girl went to her school

Referência: A garota foi à escola

Tradução: A garota foi a sua escola

B.2 – Palavra ausente (*missing words* de Popovic e Burchardt (2011)): uma palavra presente em Fonte para a qual a tradução não está presente na Tradução.

Fonte: The girl went to school

Referência: A garota foi à escola

Tradução: A garota foi escola

Neste caso é importante sempre olhar para a sentença fonte para determinar se houve remoção de alguma palavra como no exemplo a seguir no qual a Tradução está correta.

Fonte: The girl went to school

Referência: A garota foi a sua escola

Tradução: A garota foi à escola

B.3 – Palavra não traduzida (*incorreto lexical choice* de Popovic e Burchardt (2011)): uma palavra presente em Fonte que é mantida igual na Tradução, ou seja, a palavra não foi traduzida.

Fonte: The girl went to school

Referência: A garota foi à escola

Tradução: A garota foi to escola

B.4 – Palavra incorretamente traduzida (*incorreto lexical choice* de Popovic e Burchardt (2011)): uma tradução correspondente ocorreu, mas está incorreta

Fonte: The girl went to school

Referência: A garota foi à escola

Tradução: A garota foi de escola

Neste caso é importante notar que mesmo diferente da referência, se a Tradução estiver correta então não se deve marcar o erro, como ocorre no exemplo a seguir.

Fonte: The girl went to school

Referência: A garota foi à escola

Tradução: A garota foi para escola

B.5 – Palavra com erro de grafia (*incorreto lexical choice* de Popovic e Burchardt (2011)): uma tradução correspondente ocorreu na Tradução e a escolha da palavra está correta, mas sua grafia não.

Fonte: The idea was winning the game

Referência: A ideia era vencer o jogo
Tradução: A idéia era vencer o jogo

C. N-grama

- Escopo de marcação: necessariamente engloba várias palavras, senão o erro se enquadra em lexical (B).
- Descrição: o erro engloba várias palavras que formam uma expressão, seja ela semântica ou não.
- Forma de anotação: marcar as palavras da expressão em Fonte, Tradução e Referência. Na dúvida, não marcar em Fonte ou Referência.

C.1 a C.4 – Subcategorias com mesmas definições de B.1 a B.4, só que agora englobam mais de uma palavra na marcação.
Fonte: The dark horse won the game
Referência: O azarão ganhou o jogo
Tradução: O cavalo escuro ganhou o jogo

D. Ordem errada (*ordem Errada errors* de Popovic e Burchardt (2011))

- Escopo de marcação: engloba uma ou mais palavras com erros que não se enquadram nas categorias A, B e C.
- Descrição: a ordem das palavras está incorreta na Tradução, ou seja, uma sequência de uma ou mais palavras que aparece na Tradução e na Referência só que em uma posição diferente e errada em relação à que é encontrada na Referência.
- Forma de anotação: marcar da primeira até a última palavra com ordem incorreta em Fonte, Tradução e Referência. Na dúvida, não marcar em Fonte ou Referência.
  Fonte: It was the same for 1990 hosts
  Referência: Foi a mesma coisa para as anfitriãs de 1990
  Tradução: Foi a mesma coisa para as 1990 anfitriãs

3.3 Regras de anotação

Como a anotação de erros foi realizada por dois nativos do português com conhecimento satisfatório do inglês, além da especificação da tipologia de erros, também foi necessário definir um conjunto de regras de anotação que foram seguidas para assegurar as mesmas diretrizes para ambos os anotadores.

1. Anotar erros nas categorias C, D, A e B, nesta ordem[6];

2. Seguir a estratégia de anotar o mínimo necessário que precisaria ser alterado para tornar Tradução correta;

3. Não fazer a análise supondo a versão corrigida de um erro previamente anotado, por exemplo em:
   Referência: A casa do meu avô fica em São Paulo
   Tradução: A meu tio casa fica em São Paulo
   Dois erros devem ser anotados: (1) ordem incorreta (D) da sequência de palavras "meu tio casa" e (2) palavra incorretamente traduzida (B.4) "tio". Neste caso não se deve marcar o erro de palavra ausente (B2) para "do" porque se estaria supondo a correção do erro de ordem (D).

4. Podem ser anotadas várias categorias de erro na mesma sequência. Por exemplo em:
   Referência: As garotas foram à escola
   Tradução: As garoto foram à escola
   Dois erros devem ser anotados: (1) concordância de número (A.1) da palavra "garoto" e (2) concordância de gênero (A.2) da palavra "garoto".

3.4 Resultados da anotação

Dois anotadores humanos realizaram a marcação dos erros usando a ferramenta Blast (veja exemplo de anotação na seção 2.3.1.1). O processo de anotação do corpus de treinamento foi realizado em três etapas. No início, os anotadores marcaram juntos as primeiras 54 sentenças do corpus de treinamento, seguindo as regras definidas, discutindo-as e adaptando-as caso julgassem necessário. Após o passo inicial, anotaram separadamente o mesmo conjunto de 126 (sentenças 55 a 180 do mesmo corpus), com o intuito de medir a concordância entre eles, que resultou em 63% como mostra a Tabela 3.1. O cálculo da concordância entre os anotadores considerou os erros que foram marcados exatamente da mesma forma: mesma categoria e mesmas palavras em Fonte, Referência e Tradução. Nesse cálculo da concordância entre anotadores, uma sentença sem erros foi contabilizada como uma instância enquanto cada erro marcado foi considerado como uma instância. Por fim, na última etapa do processo de anotação, cada anotador marcou metade do total de sentenças restantes, chegando às 1.314 sentenças do corpus de treinamento.

Os resultados das sentenças anotadas em paralelo por ambos anotadores, por categoria de erro, são detalhados na Tabela 3.2. Nesta tabela, os valores absolutos e as porcentagens (entre

[6] A ordem foi escolhida porque em experimentos de anotação de erros de TA prévios verificou-se que algumas categorias influenciam outras e, portanto, uma ordem foi definida para otimizar o processo de anotação.

Tabela 3.1: Concordância nas anotações das mesmas sentenças realizadas em paralelo pelos dois anotadores humanos.

Sentenças consideradas	**Sentenças 55-180**			
	Primeira anotação		**Anotação após revisão**	
	Erros Iguais	**Concordância**	**Erros Iguais**	**Concordância**
Tradução	166	54%	196	67%
Fonte+Tradução	157	50%	193	65%
Fonte+Tradução+Referência	145	44%	189	63%

parênteses) de erros encontrados são apresentados nas colunas restrito, na qual o valor é dado com base nos erros marcados por ambos os anotadores; e geral, na qual o valor corresponde ao total de erros marcados por pelo menos um dos anotadores. Além disso os valores de cada anotador e sua concordância são apresentados nas colunas anotador 1, anotador 2 e concordância, respectivamente. São consideradas nos cálculos as palavras anotadas em Fonte, Tradução e Referência.

A partir dos valores desta tabela é possível notar que os erros mais comuns, anotados por ambos anotadores, são os das categorias B (erros lexicais) e A (erros sintáticos), nesta ordem. Essas categorias também tiveram, de modo geral, uma boa concordância entre anotadores.

Tabela 3.2: Erros por categoria (A, B, C, D) contendo os valores absolutos e as porcentagens correspondentes (entre parênteses): erros marcados da mesma forma por ambos os anotadores (restrito), erros marcados por pelo menos um dos anotadores (geral), erros de cada anotador (anotador 1 e anotador 2). A última coluna mostra a concordância por categoria de erro.

	SUBCATEGORIA	**Restrito**	**Geral**	**Anotador 1**	**Anotador 2**	**Concordância**
A	**concordância em número**	20(14,71)	31(12,55)	27(14,14)	24(12,5)	64,52
	concordância em gênero	16(11,76)	20(8,10)	17(8,90)	19(9,90)	80,00
	flexão verbal	19(13,97)	32(12,96)	27(14,14)	24(12,50)	59,38
	PoS	3(2,21)	6(2,43)	3(1,57)	6(3,13)	50,00
B	**palavra extra**	2(1,47)	6(2,43)	4(2,09)	4(2,08)	33,33
	palavra ausente	22(16,18)	41(16,60)	29(15,18)	34(17,71)	53,66
	palavra não traduzida	22(16,18)	27(10,93)	22(11,52)	27(14,06)	81,48
	incorretamente traduzida	20(14,71)	46(18,62)	34(17,8)	32(16,67)	43,48
	grafia	0(0,00)	5(2,02)	3(1,57)	2(1,04)	0,00
C	**n-grama ausente**	1(0,74)	1(0,40)	1(0,52)	1(0,52)	100,00
	incorretamente traduzido	3(2,21)	9(3,64)	5(2,62)	7(3,65)	33,33
D	**ordem errada**	8(5,88)	23(9,31)	19(9,95)	12(6,25)	34,78

A média de erro por sentença com erros foi de 2,46. A Tabela 3.3 traz o número total de sentenças que apresentaram de 0 a 10 erros. Das 1.314 sentenças anotadas, 33,10% estavam corretas (sem nenhum erro), ou seja, 66,90% das sentenças traduzidas pelo TAEIP na amostra analisada apresentaram um ou mais erros. Das sentenças com erros, 24,35% apresentaram apenas 1 erro, 16,29% apresentaram 2 erros, 12,18% apresentaram 3 erros e assim por diante.

Tabela 3.3: Número de sentenças traduzidas pelo TAEIP que apresentaram de 0 a 10 erros.

Erros	Sentenças
0	435
1	320
2	214
3	160
4	81
5	52
6	21
7	12
8	9
9	5
10	5

Tabela 3.4: Anotação manual por categoria de erro: quantidade de erros anotados e porcentagem.

Categoria de Erro	Subcategoria de Erro	Quantidade	%
Erros sintáticos	Concordância em número	301	14,02
	Concordância em gênero	271	12,62
	Flexão verbal	209	9,73
	PoS	48	2,24
	TOTAL	**829**	**38.61**
Erros lexicais	Palavra extra	108	5,03
	Palavra ausente	318	14,81
	Palavra não traduzida	213	9,92
	Palavra incorretamente traduzida	304	14,16
	Palavra com erro de grafia	12	0,56
	TOTAL	**955**	**44,48**
N-grama	N-grama ausente	12	0,56
	N-grama não traduzido	5	0,23
	N-grama incorretamente traduzido	153	7,13
	TOTAL	**170**	**7,92**
Reordering	Ordem	193	8,99
	TOTAL	**193**	**8,99**
TOTAL		**2.147**	**100**

Como pode ser visto na Tabela 3.4[7], os erros lexicais foram os mais frequentes (44,48%) seguidos dos erros sintáticos (38,61%). Esse resultados corroboram estudos anteriores para TA de en para pt (CASELI, 2007), nos quais os erros lexicais também ocorreram com mais frequência.

O corpus de treinamento, anotado como descrito neste capítulo, é usado para treinamento dos algoritmos de AM para identificar e corrigir erros de TA como explicado nos próximos capítulos.

[7]Das 1.314 sentenças anotadas, 126 tiveram os erros marcados paralelamente por dois anotadores. Para essas 126 sentenças, as anotações de apenas um anotador é mostrada nesta tabela pois, como a concordância entre anotadores foi considerada boa, optou-se por usar somente a de um deles no treinamento do APE.

Capítulo 4

IDENTIFICAÇÃO AUTOMÁTICA DE ERROS

Usando o corpus anotado conforme descrito no capítulo 3, foram realizados cinco experimentos para identificação automática de erros de TA. Os algoritmos de AM utilizados foram J48 (árvore de decisão), Naive Bayes e SMO (*Support Vector Machine*) disponíveis na ferramenta Weka (HALL et al., 2009). A escolha da árvore de decisão e do Naive Bayes deu-se porque são de simples interpretação, além de atingirem bons resultados em diversas aplicações. Além deles, optou-se também por testar com o SVM devido ao bom desempenho reportado em trabalhos relacionados (SPECIA, 2011).

O método de validação usado foi o *10-Fold-Cross-Validation*[1]. As *features* definidas para os experimentos são apresentadas na seção 4.1 seguidas da descrição do processo de geração das instâncias de treinamento (seção 4.2). A seção 4.3 descreve os cinco experimentos realizados para a identificação automática de erros de tradução e, por fim, a seção 4.4 discute os resultados desses experimentos.

4.1 Seleção de *features*

As *features* definidas para os experimentos desta pesquisa baseiam-se em (FISHEL et al., 2012) e (FELICE; SPECIA, 2012). Entretanto, poucas delas são exatamente as mesmas dos trabalhos relacionados.

Como definido no capítulo 3, o corpus de treinamento é composto por trios de sentenças: fonte (Src), de referência (Ref) e traduzida automaticamente (Sys). Contudo, para os experimentos descritos neste documento apenas os pares de Src e Sys são usados uma vez que Ref não estará disponível no momento da aplicação do APE resultante. Para decidir quais *features* seriam extraídas de Src e Sys, seguiu-se a diretriz de verificar onde cada uma delas pode-

[1] Validação cruzada dividindo os dados em dez conjuntos de treinamento. Alterna-se um dos conjuntos para teste com os outros nove restantes para treinamento até que o procedimento seja repetido dez vezes.

ria ajudar a identificar pelo menos uma categoria ou subcategoria de erro. Por exemplo, uma grande diferença no número de *tokens* de Src e Sys pode ser um indício de sentença incorreta, diferenças no número de gêneros entre Src e Sys poderia indicar um erro de concordância e assim por diante.

A maioria das *features* baseiam-se em uma janela de *tokens* ou *token window* (TW) que é uma sequência de *tokens* de um tamanho máximo pré-definido. Cada janela possui um *token* central (CT) e uma sequência de *tokens* antes (*tokens* precedentes) e depois (*tokens* sucessores) dele. Por exemplo, uma TW de tamanho 3 seria formada pelo CT e apenas um *token* antes e um *token* depois do CT.

A Figura 4.1 mostra exemplos de TWs de três tamanhos diferentes (cinco, sete e onze) definidos para a sentença Sys exibida na Figura 3.1

Figura 4.1: Exemplo de TWs de tamanho cinco, sete e onze, respectivamente. O CT está sublinhado.

Como nos finals do campeonato , é possível ver as duas equipes .

Como nos finals do campeonato , é possível ver as duas equipes .

Como nos finals do campeonato , é possível ver as duas equipes .

A Tabela 4.1 mostra as *features* definidas para o treinamento dos algoritmos de AM nos experimentos apresentados neste documento. Elas podem ser provenientes apenas da sentença fonte (Src), apenas da sentença traduzida (Sys), de ambas (Src e Sys) ou de uma relação entre as sentenças (Src/Sys).

Algumas das *features* presentes na Tabela 4.1 são geradas dependendo do tamanho de TW: genTokenNBefSys, genTokenNAftSys, numTokenNBefSys, numTokenNAftSys, poSTokenNBefSrc e poSTokenNAftSrc. Para essas *features*, o N representa a posição do *token* em relação ao CT. Por exemplo, se TW tiver tamanho 5 na Figura 4.1 as *features* serão: genToken1BefSys (m), genToken2BefSys (NC), genToken1AftSys (m), genToken2AftSys (m), numToken1BefSys (pl), numToken2BefSys (NC), numToken1AftSys (sg), numToken2AftSys (sg), poSToken1BefSrc (det), poSToken2BefSrc (pr), poSToken1AftSrc (pr) e poSToken2AftSrc (det).[2]

4.2 Instâncias de treinamento

Como mencionado anteriormente, as instâncias de treinamento foram definidas com base nas janelas de *tokens* (TW). Há dois tipos de TW: a TW fonte (derivada de Src) e a TW alvo

[2]Na Figura 3.2 podem ser vistos os valores dessas *features*.

Tabela 4.1: *Features* de treinamento definidas para as sentenças fonte (Src) e alvo (Sys).

Features	**Tipo**	**Origem**	**Código do Atributo**
Tamanho da sentença fonte em número de *tokens*	Numérico	Src	srcSize
Há alguma palavra na forma possessiva na TW de Src?	Boolean	Src	EngPossessive
Tamanho da sentença alvo em número de *tokens*	Numérico	Sys	sysSize
Gênero do CT da TW	String	Sys	genSysToken
Número do CT da TW	String	Sys	numSysToken
Gênero dos *tokens* que precedem o CT da TW	String	Sys	genTokenNBefSys
Gênero dos *tokens* que sucedem o CT da TW	String	Sys	genTokenNAftSys
Número dos *tokens* que precedem o CT da TW	String	Sys	numTokenNBefSys
Número dos *tokens* que sucedem o CT da TW	String	Sys	numTokenNAftSys
O CT da TW tem etiqueta morfossintática atribuída?	Boolean	Sys	sysTokenTag
O CT da TW começa com letra maiúscula?	Boolean	Sys	sysBegCap
Etiquetas morfossintáticas dos *tokens* que precedem o CT da TW	String	Src e Sys	poSTokenNBefSrc
Etiquetas morfossintáticas dos *tokens* que sucedem o CT da TW	String	Src e Sys	poSTokenNAftSrc
Etiquetas morfossintática do CT da TW	String	Src e Sys	poSSrcToken e poSSysToken
Forma+número+pessoa+tempo do CT da TW quando for um verbo	String	Src e Sys	verbSrc e verbSys
Razão do tamanho da sentença: o comprimento de Src dividido pelo comprimento de Sys	Numérico	Src/Sys	srcSysRatio
Razão de verbos: a quantidade de verbos em Src dividida pela quantidade de verbos em Sys	Numérico	Src/Sys	srcSysVRatio
Razão de substantivos: a quantidade de substantivos em Src dividida pela quantidade de substantivos em Sys	Numérico	Src/Sys	srcSysNRatio
O CT da TW de Src é o mesmo CT da TW de Sys?	Boolean	Src/Sys	equalTokenSrcSys
Há caracteres especiais, marcas de pontuação ou números nos CTs de Src e Sys?	Boolean	Src/Sys	specialCharSrcSys

(derivada de Sys). Para cada TW pode existir apenas um CT.

Nas ocorrências de erro onde apenas um *token* está anotado em Sys e somente um *token* está anotado em Src, o *token* marcado em Src é considerado como sendo o CT da TW fonte e o *token* anotado em Sys é tomado como o CT da TW alvo. Nos outros casos, deve-se definir um CT líder (LCT). O LCT define a posição do CT para as janelas de Src e também de Sys e pode vir tanto de Src como de Sys, dependendo da situação. Se o LCT vier de Src, o CT de Sys é obtido via alinhamento com Src e vice-versa. As seguintes regras definem como obter o LCT para Sys e Src:

- Quando houver um *token* anotado para o erro, ele é o LCT;
- Para dois *tokens* anotados, o primeiro é escolhido como LCT;
- Onde existir três ou mais *tokens* anotados, a posição do LCT é calculada como a parte inteira da divisão do número de *tokens* anotados por dois.[3]

[3]Por exemplo, para quatro *tokens* anotados, o LCT seria o *token* na posição 2 (4/2 = 2); no caso de três *tokens* anotados, o LCT seria o do meio, na posição 1 (3/2 = 1,5).

As regras que definem se o LCT deve vir de Src ou Sys estão abaixo, ordenadas de acordo com sua prioridade:

LCT de Sys – O LCT vem de Sys nos seguintes casos:

- A subcategoria de erro é "Palavra extra"; ou
- Há mais de um *token* anotado em Sys para uma ocorrência de erro; ou
- O *token* é aleatoriamente selecionado de uma sentença correta na geração de instâncias de treinamento.[4]

LCT de Src – O LCT é obtido de Src nas seguintes situações:

- A subcategoria de erro é "Palavra ausente" ou "N-grama ausente"; ou
- Há apenas um *token* anotado em Sys para uma ocorrência de erro e mais de um *token* anotado em Src para a mesma ocorrência.

Com base no conjunto de *features* definido, as instâncias de treinamento foram geradas a partir do corpus anotado. Para as sentenças contendo erros de TA, cada ocorrência de erro deu origem a uma instância de treinamento diferente. Cada sentença correta (sentença sem qualquer ocorrência de erro), por sua vez, produziu instâncias de treinamento aleatoriamente selecionadas representando a classe "correta".

A quantidade de instâncias usadas para o treinamento varia de acordo com o propósito do experimento, como apresentado na seção 4.3, mas todas as instâncias derivam das anotações resumidas na Tabela 3.4.

4.3 Experimentos para identificação automática de erros

Cinco experimentos foram realizados para testar os algoritmos de AM na tarefa de identificação automática de erros de TA conforme descrito nas seções a seguir.

4.3.1 Primeiro experimento: classificação em correto ou incorreto

O primeiro experimento (**E1**) fez a classificação usando duas classes: correto (sem erros) e incorreto (com pelo menos um erro). Para isso, um conjunto de treinamento balanceado composto de 4.294 instâncias foi criado: 2.147 instâncias da classe incorreta (todas as instâncias

[4] Na geração de instâncias de testes que serão checadas para verificar se há erro ou não, o LCT também vem de Sys.

de erros anotadas manualmente, veja total na Tabela 3.4) e o mesmo número de instâncias da classe correta.

Em E1, o algoritmo de árvore de decisão classificou corretamente 76,7% (precisão) das instâncias usando a janela de tamanho 5 (TW-5), já Naive Bayes e SVM atingiram suas melhores precisões no geral usando a TW-7: 73,6% e 74,5% respectivamente. O desempenho detalhado está na Tabela 4.2.

Tabela 4.2: E1 – Resultados da classificação em correto/incorreto para os classificadores Árvore de Decisão (DT), Naive Bayes (NB) e *Support Vector Machine* (SVM) para cada tamanho de janela (TW).

Tamanho de TW	Classe	Cobertura %			Precisão %		
		DT	NB	SVM	DT	NB	SVM
5	Correto	81,2	78,2	76,1	74,2	71,0	73,3
	Incorreto	71,8	68,0	72,3	79,3	75,8	75,2
	Total	**76,5**	**73,1**	**74,2**	**76,7**	**73,4**	**74,2**
7	Correto	81,5	77,7	76,2	74,0	71,6	73,7
	Incorreto	71,3	69,2	72,9	79,4	75,6	75,3
	Total	**76,4**	**73,5**	**74,5**	**76,7**	**73,6**	**74,5**
11	Correto	81,2	74,8	74,6	73,9	72,6	72,5
	Incorreto	71,4	71,7	71,7	79,2	74,0	73,8
	Total	**76,3**	**73,3**	**73,1**	**76,6**	**73,3**	**73,2**

A Figura 4.2 mostra um exemplo de erro sintático (concordância em gênero) corretamente classificado pelos três algoritmos de AM como pertencente à classe incorreto. No exemplo, o CT está sublinhado e todos os *tokens* da TW-5 aparecem em negrito.

Figura 4.2: Exemplo de erro de concordância em gênero em uma TW-5 onde o CT é a palavra com erro de concordância "*instalados*" (masculino). Como pode ser visto na sentença de referência (Ref), o correto seria "*instaladas*" (feminino). Todos os três algoritmos de AM atribuíram corretamente a classe "incorreto" à instância.

Src	Maria Lucia believes that this is an " interesting solution " , because it also prevents housing **from being installed very close** to the stream , since the space would be occupied by the treatment station .
Sys	Maria Lucia acredita que essa é uma " solução interessante " , porque também evita **moradias sejam instalados bem perto** do riacho , já que o espaço seria ocupado pela estação de tratamento .
Ref	Maria Lucia acredita que essa é uma " solução interessante " porque , além de jogar a água já tratada na bacia , também evita que as moradias sejam instaladas muito próximas ao córrego , já que o espaço seria ocupado pela estação de tratamento .

Os resultados obtidos mostram que é possível diferenciar entre correto e incorreto com aproximadamente 77% de precisão. Assim, E1 demonstrou que a identificação de erros por meio da classificação de instâncias como corretas ou incorretas pode ser usada em um APE como passo prévio à correção automática de erros de TA.

4.3.2 Segundo experimento: classificação por categoria de erro

Embora a classificação de uma instância como correta ou incorreta já seja muito útil para um APE, um novo experimento foi proposto almejando uma tarefa mais difícil: classificar por categorias de erro. No segundo experimento (**E2**), o mesmo conjunto de *features* e algoritmos de AM foram aplicados para tentar uma classificação mais detalhada. A tarefa de classificação envolve 5 classes: as quatro categorias de erros apresentadas na seção 3.2 (erros sintáticos, erros lexicais, n-grama ou ordem errada), além da classe "correto" (sem erros).

O conjunto de treinamento gerado para este experimento contém 2.684 instâncias e a classe a ser atribuída é a categoria de erro ou "correto". Para preservar o equilíbrio entre o número de instâncias de cada classe, neste experimento, a quantidade de instâncias para a classe "correto" corresponde ao número total de instâncias representando erros (2.147) dividido pelo número de categorias de erros (4). Portanto, o conjunto de treinamento de E2 está composto por: 955 instâncias para a classe "erros lexicais", 829 para "erros sintáticos", 193 para "ordem errada" e 170 para "n-grama" e, finalmente, 537 instâncias para a classe "correto".

Na Tabela 4.3 estão os resultados de E2. O melhor desempenho geral foi obtido novamente pelo algoritmo árvore de decisão com TW-5. Como esperado, os resultados de E2 não foram tão bons como os de E1. Entretanto, é possível notar, na Tabela 4.3, que, com poucas instâncias de treinamento, as *features* utilizadas e os algoritmos de AM escolhidos obtiveram uma cobertura de 61% e uma precisão de 58% no geral. Na categoria de erro "erros sintáticos" obteve-se cerca de 81% de cobertura e 65% de precisão, e nos erros lexicais, 72% de cobertura e 67% de precisão. Conclui-se, então, que bons resultados foram obtidos na classificação para as duas categorias de erros com a maior quantidade de instâncias de treinamento.

A Figura 4.3 traz um exemplo de erro de ordem corretamente classificado por DT e NB, mas não por SVM. O CT está sublinhado e todos os *tokens* que compõem TW-7 estão em negrito para o exemplo.

Embora a precisão geral obtida pelo melhor algoritmo (DT) em E2, 57%, seja cerca de 20 pontos percentuais menor do que a de E1, é importante mencionar que para as categorias de erros com as maiores quantidades de instâncias (erros lexicais e erros sintáticos), essa diferença cai para cerca de 10 pontos percentuais. Esse é um indício de que a quantidade de instâncias de treinamento influencia bastante o aprendizado desta tarefa.

4.3.3 Terceiro experimento: classificação para subcategorias de erros

Tendo como objetivo testar uma classificação ainda mais detalhada, um terceiro experimento (**E3**) foi organizado, ainda usando as mesmas *features* e algoritmos de AM dos anterio-

Tabela 4.3: E2 – Resultados de classificação por categoria de erro para os classificadores Naive Bayes (NB), Árvore de Decisão (DT) e *Support Vector Machine* (SVM) para cada tamanho de TW. As categorias de erro estão ordenadas pelo número de instâncias de treinamento em ordem descrescente.

Tamanho de TW	Classe	Cobertura %			Precisão %		
		DT	NB	SVM	DT	NB	SVM
5	Erros lexicais	71,7	50,1	67,7	67,1	67,7	63,9
	Erros sintáticos	81,5	74,8	70,8	64,8	63,5	67,4
	Sem erros	47,7	46,7	53,4	52,0	42,7	50,3
	Ordem Errada	14,5	41,5	22,3	38,9	22,2	30,5
	N-grama	4,1	5,3	7,1	12,7	17,0	13,5
	Total	**61,5**	**53,6**	**58,7**	**57,9**	**54,9**	**56,7**
7	Erros lexicais	72,1	45,8	66,3	66,6	65,6	62,8
	Erros sintáticos	81,3	75,4	67,3	64,6	60,1	68,8
	Sem erros	45,6	41,0	53,1	51,8	41,3	48,7
	Ordem Errada	11,9	43,5	21,2	31,1	21,3	30,8
	N-grama	4,7	4,7	12,4	13,3	16,0	14,3
	Total	**61,1**	**51,2**	**57,3**	**57,1**	**52,7**	**56,5**
11	Erros lexicais	72,1	40,8	60,8	66,6	59,2	61,7
	Erros sintáticos	81,1	73,8	63,9	64,0	57,5	69,2
	Sem erros	45,8	33,0	49,9	52,3	37,6	41,2
	Ordem Errada	13,5	43,0	21,8	34,2	18,2	29,6
	N-grama	7,1	1,8	11,8	22,2	9,1	10,9
	Total	**61,3**	**47,1**	**53,7**	**57,8**	**48,2**	**54,4**

Figura 4.3: Exemplo de uma ocorrência de erro de ordem em uma TW-7 onde o CT é a palavra "mercado" (*market*). Como mostrado na sentença de referência (Ref), a tradução correta para o segmento "*market strategies*" deveria ser "estratégias de mercado" e não "mercado estratégias". Os classificadores DT e NB classificaram essa instância corretamente como ordem errada, enquanto o SVM a classificou como correta (sem erros).

Src	Paschoal explains that some companies are at a stage he classifies as " pre - competitive " , as he says ; that is , in order to perfect the products , they need knowledge of general interest that does not **interfere with the market strategies of the** sector as a whole .
Sys	Paschoal explica que algumas empresas estão num estágio classifica como " pré - competitivo " , como ele diz , ou seja , para aperfeiçoar os produtos precisam saber do interesse geral que **não interfere no mercado estratégias do setor** como um todo .
Ref	Paschoal explica que algumas empresas estão em estágio que qualifica de " pré - competitivo " , ou seja , para aperfeiçoar os produtos , necessitam de conhecimentos de interesse geral que não interferem nas estratégias de mercado do conjunto do setor .

res. Desta vez são 14 classes possíveis: as 13 subcategorias de erro (veja Tabela 3.4) além da classe "correto" (sem erros). O conjunto de treinamento foi gerado seguindo a mesma estratégia de balanceamento de E2 resultando em 2.312 instâncias: 165 instâncias para a classe "correto" (2.147 dividido por 13) e a quantidade de instâncias na Tabela 3.4 para cada subcategoria de erro.

A Tabela 4.4 mostra os resultados para NB, DT e SVM de E3. Os resultados gerais de cobertura variam de 36,37% usando SVM e TW-11 até 44,20% usando Naive Bayes (NB) e TW-5.

Embora os resultados gerais não sejam bons, a subcategoria "Palavra não traduzida" apresentou precisão entre 70% e 80% e "Palavra ausente" obteve uma precisão acima de 60% para os três algoritmos de AM e os três diferentes tamanhos de TW. Todas as outras subcategorias ficaram abaixo de 50% nesse experimento. A "Palavra ausente" teve o segundo melhor desempenho provavelmente porque a palavra na sentença Src tende a não estar alinhada com qualquer palavra em Sys quando a palavra for ausente em Sys. Isso faz com que as *features* baseadas em Sys fiquem vazias (valor NULL). Já para "Palavra não traduzida", subcategoria que obteve o melhor desempenho, uma *feature* que pode indicar o problema precisamente é a etiqueta PoS que recebe o valor "NC" do etiquetador sempre que uma palavra não é reconhecida como sendo parte da língua sendo processada.

Vale a pena mencionar que o tamanho pequeno do corpus de treinamento teve um grande impacto nos resultados de E3, pois algumas subcategorias tinham, por exemplo, apenas 5 (n-grama não traduzido) ou 12 (palavras com erro de grafia e n-gram ausente) instâncias de treinamento.

Na Figura 4.4 há um exemplo de erro de "palavra não traduzida" corretamente classificado pelos três algoritmos de AM. O CT está sublinhado e todos os *tokens* que fazem parte de TW-5 estão em negrito.

Figura 4.4: Exemplo de uma ocorrência de "palavra não traduzida" em uma TW-5 onde CT é a palavra não traduzida "horn". Como mostrado pela sentença de referência (Ref), a tradução correta para a palavra deveria ser "trompa". Todos os três algoritmos de AM classificaram corretamente a instância como um erro de "palavra não traduzida"

Src	At the beginning of their careers as musicians , Iazzetta studied percussion , e Ferraz **, the horn .**
Sys	No início de sua carreira como músicos , Iazzetta estudou percussão e **Ferraz , horn .**
Ref	No início de suas carreiras como músicos , Iazzetta estudou percussão e Ferraz , trompa .

4.3.4 Quarto experimento: classificação em dois passos (E1→E2)

Com o intuito de verificar se a precisão encontrada em E2 poderia ser melhorada ao realizar a classificação por categoria de erro em duas etapas (E1→E2), o quarto experimento (**E4**) foi planejado. Assim, E4 foi realizado considerando-se a combinação de algoritmo de AM e TW que apresentou a melhor precisão nos experimentos E1 e E2: árvore de decisão (DT) e TW-5.

Para tanto, primeiramente as instâncias foram classificadas em correto/incorreto pelo classificador treinado em E1. Em seguinda, as instâncias foram enviadas ao classificador por categoria de erro treinado em E2. Os resultados de E4 estão na Tabela 4.5.[5]

[5]A classe "sem erros" da Tabela 4.5 corresponde às instâncias classificadas incorretamente como "incorreto"

Tabela 4.4: E3 – Resultados de classificação por subcategoria de erro para os classificadores Naive Bayes (NB), Árvore de Decisão (DT) e *Support Vector Machine* (SVM) para cada tamanho de TW. As subcategorias de erro são ordenadas pelo número de instâncias de treinamento em ordem descrescente.

Tamanho de W	Classe	Cobertura %			Precisão %		
		DT	NB	SVM	DT	NB	SVM
5	Palavra ausente	72,6	70,8	68,2	64,2	68,2	63,3
	Palavra incorretamente traduzida	27,6	28,3	25,7	26,9	29,8	25,0
	Concordância em número	32,2	29,9	32,6	31,8	35,6	34,8
	Concordância em gênero	46,1	52,8	50,9	39,7	45,0	46,9
	Palavra não traduzida	89,7	88,3	80,3	79,9	72,3	77,0
	Flexão verbal	46,9	55,5	38,8	47,6	40,3	42,4
	Ordem errada	20,2	43,5	23,3	27,5	29,1	29,4
	Sem erros	29,7	15,8	25,5	28,2	34,2	21,3
	N-grama incorretamente traduzido	12,4	6,5	15,7	13,8	17,2	14,3
	Palavra extra	25,9	50,0	33,3	29,8	36,7	31,9
	PoS	10,4	0,0	4,2	23,8	0,0	11,8
	Palavra com erro de grafia	0,0	0,0	8,3	0,0	0,0	9,1
	N-grama ausente	0,0	0,0	0,0	0,0	0,0	0,0
	N-grama não traduzido	0,0	0,0	0,0	0,0	0,0	0,0
	Total	**41,8**	**44,2**	**40,4**	**39,9**	**41,2**	**39,6**
7	Palavra ausente	73,0	70,8	68,6	64,3	64,3	62,6
	Palavra incorretamente traduzida	25,7	29,3	20,1	25,7	27,4	20,5
	Concordância em número	32,2	29,9	27,6	31,6	35,3	30,7
	Concordância em gênero	45,4	52,8	45,0	39,4	44,7	42,5
	Palavra não traduzida	90,1	88,3	77,5	79,7	72,6	77,1
	Flexão verbal	45,9	55,5	38,8	46,6	41,3	44,3
	Ordem errada	18,1	43,0	21,2	27,6	31,0	27,2
	Sem erros	29,7	11,5	31,5	25,8	28,8	25,0
	N-grama incorretamente traduzido	12,4	5,9	18,3	14,0	16,7	14,9
	Palavra extra	24,1	41,7	26,9	25,0	34,4	25,4
	PoS	12,5	0,0	14,6	31,6	0,0	21,2
	Palavra com erro de grafia	0,0	0,0	0,0	0,0	0,0	0,0
	N-grama ausente	0,0	0,0	8,3	0,0	0,0	10,0
	N-grama não traduzido	0,0	0,0	0,0	0,0	0,0	0,0
	Total	**41,2**	**43,6**	**38,4**	**39,4**	**40,0**	**38,1**
11	Palavra ausente	73,3	71,7	66,4	63,5	61,3	60,1
	Palavra incorretamente traduzida	24,0	26,3	16,8	25,6	22,2	19,0
	Concordância em número	29,6	27,9	29,2	33,1	34,0	30,1
	Concordância em gênero	47,2	49,1	38,4	38,2	43,3	37,8
	Palavra não traduzida	90,1	81,7	76,5	79,0	70,2	76,5
	Flexão verbal	43,5	50,2	33,5	42,7	37,9	40,2
	Ordem errada	20,2	38,3	24,9	28,5	28,9	32,0
	Sem erros	30,9	8,5	22,4	25,8	20,3	17,2
	N-grama incorretamente traduzido	15,7	6,5	18,3	17,0	19,6	14,4
	Palavra extra	21,3	39,8	32,4	24,5	34,7	27,8
	PoS	10,4	0,0	6,3	20,0	0,0	9,7
	Palavra com erro de grafia	0,0	0,0	25,0	0,0	0,0	37,5
	N-grama ausente	0,0	0,0	0,0	0,0	0,0	0,0
	N-grama não traduzido	0,0	0,0	0,0	0,0	0,0	0,0
	Total	**41,0**	**40,9**	**36,4**	**38,9**	**37,5**	**36,4**

As únicas duas categorias de erro que apresentaram melhora em E4 em comparação com os resultados de E2 foram “erros lexicais” – uma melhora de 7 pontos percentuais em cobertura

no primeiro passo (E1).

Tabela 4.5: E4 – Resultados de classificação por categoria de erro em dois passos (E1→E2) para o classificador árvore de decisão (DT) usando uma TW-5. As categorias de erro estão ordenadas pelo número de instâncias de treinamento geradas no primeiro passo, em ordem descrescente.

Categoria de Erro	Número de instâncias no treinamento	Cobertura %	Precisão %
Erros lexicais	794	78,7	70,4
Erros sintáticos	738	83,6	60,5
Sem erros	279	8,2	32,9
N-grama	121	3,3	13,8
Ordem Errada	109	6,4	20,6
Total	**2041**	**62,5**	**55,7**

(de 72% em E2 para 79% em E4) e uma melhora de 3 pontos percentuais em precisão (de 67% em E2 para 70% em E4) – e "erros sintáticos" – uma melhora de 2 pontos percentuais em cobertura de 81,5% em E2 para 83,6% em E4), mas houve perda na precisão (de 65% para 60,5%). Novamente, as categorias que apresentaram melhor desempenho foram as que possuem maior número de instâncias de treinamento o que indica, outra vez, que o desempenho ruim nas demais categorias pode ser explicado pelo pequeno número de instâncias.

4.3.5 Quinto experimento: avaliação manual de instâncias automaticamente classificadas no corpus de teste (E5)

O melhor desempenho em termos de precisão na classificação de instâncias entre correto e incorreto em E1 foi obtido pelo algoritmo de árvore de decisão (DT) usando TW-5 e TW-7. A fim de avaliar o desempenho desses modelos treinados em E1 em um novo corpus, executou-se um quinto experimento (**E5**).

Em E5, os modelos treinados em E1 usando o corpus de treinamento (test-a da FAPESP) foram aplicados ao corpus test-b da FAPESP.[6] Em seguida, 100 sentenças selecionadas aleatoriamente foram verificadas manualmente para cada TW. A Tabela 4.6 traz os valores de precisão e cobertura (na avaliação feita por um humano) dos modelos treinados.

Tabela 4.6: E5 – Resultados da avaliação manual da classificação em correto/incorreto de instâncias do corpus de teste usando árvore de decisão (DT) e TW-5/ TW-7.

Tamanho de TW	Classe	Cobertura %	Precisão %	Número de instâncias verificadas
5	Correto	60,9	61,9	64
	Incorreto	33,3	32,4	36
	Total	**50,0**	**50,0**	**100**
7	Correto	54,7	49,2	53
	Incorreto	36,2	41,5	47
	Total	**46,0**	**46,0**	**100**

[6] O corpus test-b da FAPESP está disponível em: <http://pers-www.wlv.ac.uk/∼in1676/resources/fapesp/index.html>. Acesso em: 21 jan. 2014. As sentenças Src do test-b foram traduzidas pelo TAEIP.

4.4 Discussão dos resultados para identificação automática de erros

Os experimentos relatados neste documento mostram que é possível conseguir boa precisão (77%) na classificação de um segmento traduzido por TA como correto ou incorreto (E1) com base em uma TW de tamanho 5 e um conjunto pequeno de 32 *features*.[7] Contudo, esse treinamento ainda precisa ser melhorado, pois com o corpus de teste contendo instâncias diferentes das usadas no treinamento a precisão cai para 62% (conforme descrito na seção 4.3.5). Uma possível solução para esse problema é aumentar o corpus de treinamento na tentativa de cobrir a diversidade de erros nas instâncias.

Usando o mesmo algoritmo de AM e tamanho de TW de E1, o segundo experimento (E2) mostrou um bom desempenho ao classificar os erros sintáticos (cobertura de 81% e precisão de 65%) e erros lexicais (cobertura de 72% e precisão de 67%), mas mostrou resultados ruins para as demais categorias de erro. O fato de as melhores taxas de classificação terem sido obtidas para as categorias de erro mais frequentes – de acordo com a Tabela 3.4 os erros lexicais representam 44,48% e os sintáticos 38,61% do total de erros –, indica que a abordagem proposta pode levar a bons resultados na pós-edição automática desde que existam instâncias suficientes para o treinamento dos algoritmos de AM.

Os valores baixos na classificação de algumas categorias de erros, obtidos em E2, e na maioria das subcategorias de E3, mostram que mais instâncias de treinamento seriam necessárias para melhorar a identificação por categorias e subcategorias de erros. Outro fator é a sobreposição de erros atribuídos para uma mesma TW, o que pode explicar alguns resultados ruins, pois insere ruído no corpus de treinamento. Para se ter uma ideia do impacto dessa sobreposição de categorias, a Tabela 4.7 traz a quantidade e porcentagem de ocorrências de erro com mais de um erro anotado por categoria e subcategoria.

Os experimentos com identificação automática também obtiveram resultados interessantes ao classificar as classes de erros sintáticos e erros lexicais em duas etapas (E1→E2) no quarto experimento (E4). Nesse caso, a cobertura na classificação de erros lexicais aumentou 7 pontos percentuais e a precisão aumentou 3 pontos percentuais, nos erros sintáticos a cobertura aumentou 2 pontos percentuais, embora tenha havido perda na precisão. Veja que o ganho de desempenho ocorreu mesmo com menos instâncias de treinamento disponíveis devido às instâncias incorretamente classificadas na primeira etapa (E1).

Além do aumento do corpus de treinamento, outras *features* podem ser investigadas em experimentos futuros para a identificação de erros como:

[7] Como explicado na seção 4.1, o número de *features* varia dependendo do tamanho da TW. São 32 *features* para TW-5, 40 para TW-7 e 56 para TW-11.

Tabela 4.7: Quantidade e porcentagem de ocorrências de erro com mais de um erro por categoria e subcategoria de erro.

Categoria de Erro	Subcategoria de Erro	Quantidade	%
Erros sintáticos	Concordância em número	70	23,26
	Concordância em gênero	66	24,35
	Flexão verbal	14	6,70
	PoS	1	2,08
	TOTAL	**151**	**18,21**
Erros lexicais	Palavra extra	1	0,93
	Palavra ausente	5	1,57
	Palavra não traduzida	15	7,04
	Palavra incorretamente traduzida	27	8,88
	Palavra com erro de grafia	0	0,00
	TOTAL	**48**	**5,02**
N-grama	N-grama ausente	1	8,33
	N-grama não traduzido	0	0,00
	N-grama incorretamente traduzido	5	3,28
	TOTAL	**6**	**3,53**
Ordem Errada	Ordem Errada	57	29,53
	TOTAL	**57**	**29,53**

- Uso de características obtidas a partir de árvores semânticas rasas (*shallow semantic trees*) descritas em (AZIZ; RIOS; SPECIA, 2011);
- Uso de relações semânticas como hiponímia e meronímia presentes nas instâncias (TABA; CASELI, 2012);
- Uso de características observáveis em árvores sintáticas da tradução em comparação com árvores sintáticas da sentença fonte (FELICE; SPECIA, 2012).

Com base nos resultados dos experimentos apresentados neste capítulo, acredita-se que as taxas de classificação obtidas permitem que a configuração DT+TW-5 envie ao módulo de correção, com um certo grau de confiança, as instâncias classificadas como incorretas ou, em uma classificação mais refinada, os erros sintáticos e os erros lexicais.

Capítulo 5

CORREÇÃO AUTOMÁTICA DE ERROS

Além da identificação automática de erros produzidos na TA, nesta pesquisa também investigou-se a etapa principal do APE: a correção automática desses erros. Este capítulo trata dessa etapa conclusiva.

Para o aprendizado das regras de correção automática optou-se pelo uso da técnica de AM TBL (BRILL, 1995), por meio da ferramenta μ-TBL (LAGER, 1999)[1]. Tal ferramenta foi selecionada porque, de acordo com Elming (2006), ela se mostrou bastante útil na extração de regras de pós-edição de erros de TA. A ferramenta, implementada em Prolog, facilita o uso de TBL uma vez que usa uma forma generalizada que emprega a interpretação lógica de regras de transformação, e isso traz flexibilidade e agilidade na fase de treinamento.

Assim, o APE criado para a pós-edição automática de erros de TA de inglês para o português do Brasil, Editor de Traduções Automáticas , de agora em diante referenciado pela sigla EdiTA, pode ser executado para desempenhar diversos tipos de tarefas parametrizáveis. São elas:

- Gerar arquivos de treinamento para o identificador automático de erros no formato exigido pela ferramenta Weka (veja capítulo 4);
- Classificar segmentos de 5, 7 ou 11 *tokens* de sentenças traduzidas em correto ou incorreto, ou ainda segundo sua categoria ou subcategoria de erro (veja capítulo 4);
- Criar arquivos de treinamento para o corretor automático de erros no formato exigido pela ferramenta μ-TBL (veja seção 5.1);
- Interpretar as regras geradas pela μ-TBL (veja seção 5.3);
- Aplicar as regras de correção usando o filtro do identificador automático de erros, isto é, apenas segmentos de 5, 7 ou 11 *tokens* identificados como incorretos ou que apresentem

[1] Disponível em: <http://www.ling.gu.se/∼lager/mutbl.html>. Acesso em: 27 nov. 2013.

alguma categoria ou subcategoria de erro são pós-editados (veja seção 5.4). Esse tipo de pós-edição é referenciado neste documento como "pós-edição com filtro";

- Aplicar as regras de correção nas sentenças traduzidas por TA sem passar pela fase de identificação de erros (veja seção 5.4). Esse tipo de pós-edição é referenciado neste documento como "pós-edição direta".

Após a geração dos dados de treinamento, descrita na seção 5.1, as regras são aprendidas usando a μ-TBL conforme descrito na seção 5.2. Em seguida as regras são interpretadas e aplicadas conforme descrito nas seções 5.3 e 5.4. Antes da apresentação dos experimentos na seção 5.6, a seção 5.5 descreve o tradutor automático estatístico baseado em frases TAEIP usado como *baseline* nesta pesquisa. Por fim, a seção 5.7 discute os resultados obtidos e apresenta sugestões para novos experimentos.

5.1 Geração dos dados de treinamento

Uma das tarefas desempenhadas pelo EdiTA é a geração dos dados de treinamento no formato de entrada da μ-TBL. O corpus de treinamento anotado com os erros de tradução automática e os dados de alinhamento e etiquetas morfossintáticas (veja capítulo 3, Figura 3.2) foram lidos e processados pelo EdiTA para se chegar à representação de dados no padrão necessário.

Enquanto nos experimentos com identificação automática (seção 4.3) foram desconsideradas as sentenças traduzidas por humanos (Ref) e as *features* foram extraídas somente das sentenças em língua fonte (Src) e das sentenças traduzidas por TA (Sys); no aprendizado de regras de correção usou-se Ref e Sys, sendo que Src foi desconsiderado.

Para permitir o aprendizado das regras de correção usando a μ-TBL, os dados obtidos de Sys e Ref são representados em cláusulas[2] onde os tipos de predicado[3] fazem referência às *features*. Há dois tipos de cláusula, aqui denominadas T1 e T2:

T1 (*static features* (LAGER, 2000)) – expressa a informação que está presente em determinada posição de um *token*. É derivada de Sys e podem existir várias cláusulas T1 para o mesmo *token*. As cláusulas deste tipo servem para guiar o aprendizado, pois contêm informações relevantes sobre um determinado *token*, indexadas pela posição do *token*. Esses dados, e sua posição, são usados na composição de uma regra. Exemplos destas cláusulas (para o *token* presente na posição 427) podem ser vistos na Figura 5.2.

[2] O termo vem da linguagem de programação Prolog, baseada na Lógica de Predicados, que recebe suas instruções em forma de cláusulas.

[3] Na Lógica de Predicados, um predicado é uma declaração que pode assumir os valores "verdadeiro" ou "falso".

T2 (*dynamic features* (LAGER, 2000)) – representa a informação presente em determinada posição de um *token* em Sys e seu equivalente em Ref: a mesma informação, caso ela esteja correta, ou a informação correta (em Ref) que deve substituir a atual (em Sys), no caso de erro. Veja que a informação que compõe T2 é obtida através do alinhamento do *token* em Sys com o *token* em Ref somente para *tokens* marcados com algum erro no corpus anotado. Se não houver erro anotado ou o *token* anotado em Sys não estiver alinhado com Ref, a informação em Sys é repetida onde constaria a informação vinda de Ref. As cláusulas deste tipo servem para definir o aprendizado, pois contêm informações específicas para a correção relativa ao tipo de erro que se deseja tratar, indexadas pela posição do *token*. Diferente do que ocorre com as cláusulas T1, no arquivo de treinamento usado para o aprendizado de regras que corrigem um tipo de erro, pode haver somente uma cláusula T2 para cada *token*. Exemplos destas cláusulas (para o *token* da posição 427) podem ser vistos na Figura 5.3.

Há 5 tipos de arquivos de treinamento, um para cada tipo de erro a ser corrigido. Essa divisão em vários arquivos de treinamento é uma limitação da ferramenta μ-TBL. Na versão aplicada nos experimentos deste documento (versão 1.0), de acordo com o indicado em (LAGER, 2000), para cada *token* a ferramenta aceita diferentes tipos de cláusula T1, mas apenas uma cláusula do tipo T2. Portanto, nos arquivos de treinamento, enquanto as cláusulas T1 são as mesmas em todos eles, as cláusulas T2 trazem informações somente sobre o tipo de erro que se deseja aprender a corrigir. Para os experimentos descritos neste capítulo, os arquivos foram divididos de acordo com o tipo de erro que se deseja tratar em:

- gener: concordância em gênero;
- nume: concordância em número;
- pofs: mudança de PoS;
- verboc: flexão verbal;
- wd: erros de tradução que não se encaixem nos itens acima, principalmente palavras extras e erros de ordenação.

É importante notar que os tipos de erros tratados na etapa de correção são mais específicos do que as categorias de erros apresentadas nos capítulos 3 e 4. Esses tipos, na verdade, equivalem a algumas subcategorias de erros do capítulo 3: gener (A.2), nume (A.1), pofs (A.4), verboc (A.3) e wd (B.1, B.3, B.4, B.5, C e D). Apesar de serem considerados em wd, os erros das subcategorias B.3 (palavra não traduzida), B.4 (palavra incorretamente traduzida) e B.5 (palavra com erro de grafia) só poderiam ser tratados de forma específica com o uso de recursos

extras como bases lexicais e, por isso, não houve alterações registradas para os referidos tipos nesta primeira investigação.

Os erros da subcategoria B.2 (palavra ausente) não foram tratados, pois requeririam um processamento diferenciado a fim de verificar qual palavra em Src – alinhada com palavras nas imediações da posição onde houve indicação de ausência de *token* em Sys – não possui alinhamento com qualquer palavra em Sys. Além disso, uma vez encontrada a provável palavra ausente em Src, um dicionário bilíngue seria necessário para traduzir a palavra antes de inserí-la em Sys. Essa funcionalidade não foi incluída na versão sendo discutida. É importante ressaltar também que os erros da categoria C (n-grama) e D (ordem) apresentam grande complexidade quando é preciso determinar a melhor posição dos *tokens* na substituição. Como a substituição do EdiTA é feita *token* a *token* nesta versão, esses tipos de erro foram tratados parcialmente.

A decisão de não seguir exatamente a mesma categorização definida para a anotação manual e adotada na etapa de identificação de erros foi tomada com base em dois fatos: (i) as regras aprendidas devem ser compatíveis com as categorias de erros identificadas, mas não dependentes delas já que se pretende analisar o desempenho do EdiTA com e sem a etapa de identificação de erros e (ii) a ferramenta μ-TBL tem a limitação de só aplicar uma alteração por vez, ou seja, apenas uma cláusula do tipo T2 é permitida em um arquivo de treinamento para um determinado *token*.

Assim, cada arquivo de treinamento contém cláusulas com informação de gênero (gener), número (nume), etiqueta PoS (pofs), flexão verbal (verboc)[4], forma superficial da palavra (wd) e tipo de erro (err)[5]. A Figura 5.1 traz um exemplo de erro de concordância na palavra em destaque. As cláusulas T2 para a palavra com erro mostrada no exemplo estão na Figura 5.3 como aparecem em cada tipo de arquivo. Ainda seguindo o mesmo exemplo, as cláusulas T1 podem ser vistas na Figura 5.2 e se repetem em todos os arquivos.

Figura 5.1: Exemplo de erro de concordância em gênero e número para a palavra em destaque (sublinhada).

Src	But there is one part of this research whose results can already be **adopted** .
Sys	Mas há uma parte dessa pesquisa cujos resultados já podem ser **adotada** .
Ref	Mas há uma vertente dessa pesquisa cujos resultados já podem ser **adotados** .

Pode-se verificar na Figura 5.3 que na cláusula T2 do tipo pofs ‘v’ é repetido tanto na posição referente a Sys como a Ref. Isso se dá porque não houve erro de mudança de PoS no referido exemplo. Quando trata-se de erros do tipo “n-grama ausente” ou “palavra ausente”, de

[4]Os dados sobre flexão verbal presentes no arquivo de treinamento do μ-TBL são: forma, número, pessoa, tempo e são separados por ‘.’. ‘NApp’ indica que a informação não é aplicável.

[5]O tipo de erro nessa versão não é usado para o aprendizado e tem caráter meramente informativo. Se mais de um tipo de erro existir para a mesma palavra, o primeiro que foi anotado é selecionado.

Figura 5.2: Exemplo de cláusulas T1 presentes nos 5 tipos de arquivos de entrada da ferramenta μ-TBL para o exemplo da Figura 5.1.

Tipo de arquivo	Cláusula T1
Todos	gener(427,f). nume(427,sg). verboc(427,'pp.sg.NApp.NApp'). wd(427,adotada). err(427,morph).

Figura 5.3: Exemplo de cláusulas T2 para cada um dos 5 tipos de arquivos de entrada da ferramenta μ-TBL para o exemplo da Figura 5.1.

Tipo de arquivo	Cláusula T2
gener	gener(f,m,427).
nume	nume(sg,pl,427).
pofs	pofs(v,v,427).
verboc	verboc('pp.sg.NApp.NApp','pp.pl.NApp.NApp',427).
wd	wd(adotada,adotados,427).

acordo com o padrão da ferramenta, o caracter 0 é colocado em T1 e T2 no lugar onde deveriam estar os dados de Sys sobre a palavra. Já quando é um erro de "palavra extra" o caracter 0 é colocado em T2 onde deveriam estar os dados de Ref.

Para cada tipo de arquivo de treinamento (gener, nume, pofs, verboc, wd) há um arquivo auxiliar chamado de arquivo de *templates*. Esses arquivos auxiliares são necessários para o aprendizado usando TBL, já que as regras aprendidas são, na verdade, instâncias de *templates*. Os *templates* têm o mesmo formato de regras, no entanto, usam variáveis no lugar de valores (LAGER, 2000). Os *templates* criados para os experimentos levam em conta a influência contextual de dados de no máximo quatro *tokens* antes e depois do *token* de interesse. O número quatro ficou definido porque nos *templates* criados manualmente não foi preciso criar qualquer *template* que verificasse um *token* mais distante do que quatro posições (antes ou depois) do *token* sendo analisado. Além disso, os arquivos de *templates* usados como modelo tampouco traziam verificações mais distantes do que quatro posições do *token* de interesse.

Para a correção dos erros de concordância em gênero e número mostrados no exemplo da Figura 5.1, os dois *templates* da Figura 5.4 poderiam ser instanciados gerando, por exemplo, as regras mostradas na Figura 5.5. As regras aprendidas são automaticamente gravadas pela ferramenta μ-TBL em um arquivo de saída.

Na Figura 5.4, o *template* referente a gênero (gener) indica que o gênero de um *token* seja substituído por outro ("gener:_>_") se ("<-") determinado valor estiver presente no gênero

Figura 5.4: Exemplos de *templates* a serem instanciados para correção dos erros da Figura 5.1.

Tipo de arquivo auxiliar	Exemplo de *template*
gener	gener:_>_ <- gener:_@[-1,-2,-3,-4].
nume	nume: _>_ <- nume:_@[-1,-2] & verboc:_@[0].

Figura 5.5: Exemplos de possíveis regras geradas a partir dos *templates* da Figura 5.4.

Tipo de regra	Exemplo de regra
gener	gener:f>m <- gener:m@[-1,-2,-3,-4]
nume	nume:sg>pl <- nume:pl@[-1,-2] & verboc:'pp.sg.NApp.NApp'@[0]

("gener:_") de pelo menos um dos quatro *tokens* que o precedem ("@[-1,-2,-3,-4]") . Já o *template* referente a número (nume) indica que o número de um *token* seja substituído por outro ("nume:_>_") se ("<-") determinado valor estiver presente no número ("nume:_") de pelo menos um dos dois *tokens* que o precedem ("@[-1,-2]") e ("&") houver determinado valor nos dados de verboc (verboc:_) do *token* sendo analisado ("@[0]") . Como pode-se notar, o símbolo "_" indica variáveis a serem instanciadas e os números indicam a posição relativa dos demais *tokens* ao *token* sendo analisado (sempre indicado como presente na posição 0).

A Figura 5.5 contém as regras correspondentes aos *templates* da Figura 5.4 com suas variáveis já instanciadas. A regra referente a gênero (gener) indica que o gênero feminino de um *token* seja substituído pelo masculino ("gener:f>m") se ("<-") o valor masculino estiver presente no gênero ("gener:m") de pelo menos um dos quatro *tokens* que o precedem ("@[-1,-2,-3,-4]"). Já a regra referente a número (nume) indica que o singular seja substituído por plural ("nume:sg>pl") se ("<-") o número do *token* for plural ("nume:pl") em pelo menos um dos dois *tokens* que o precedem ("@[-1,-2]") e ("&") as informações de verbo indicarem particípio e singular (verboc:'pp.sg.NApp.NApp') no *token* sendo analisado ("@[0]") .

5.2 Aprendizado automático de regras de correção

Como mencionado no capítulo 2, TBL (*Transformation-based Learning*) é uma técnica de AM que aprende regras de correção por meio da busca pela regra que causa a maior redução de erro em um determinado passo do treinamento. A regra encontrada é, então, aplicada aos dados de treinamento para a correção dos erros e o processo de busca recomeça. Esse processo de busca-aplicação de regras se repete até que a diminuição dos erros atinja um nível abaixo do definido previamente. O resultado deste processo é um conjunto de regras e a prioridade na qual devem ser aplicadas. Idealmente esse conjunto de regras reduz os erros ao máximo.

Para que o aprendizado de regras usando a ferramenta μ-TBL seja possível é preciso que

os seguintes parâmetros sejam especificados:

- Dados de treinamento: nome e localização do arquivo contendo os dados de treinamento no formato apresentado na seção 5.1.

- Dados de teste: nome e localização do arquivo contendo os dados de teste (no mesmo formato dos dados de treinamento);

- *Templates*: nome e localização do arquivo contendo os *templates* (detalhes sobre *templates* são apresentados na seção 5.1);

- Precisão (*accuracy*): valor a ser considerado para a precisão de uma regra (R), ou seja, sua porcentagem de acertos calculada com base no número de instâncias positivas (pos) e negativas (neg) geradas pela regra:

$$accuracy(R) = \frac{|pos(R)|}{|pos(R)| + |neg(R)|} \tag{5.1}$$

- Pontuação (*score*): número absoluto de acertos, ou seja, instâncias positivas (pos) geradas por uma regra (R) a mais do que instâncias negativas (neg) geradas por ela :

$$score(R) = |pos(R)| - |neg(R)| \tag{5.2}$$

Com os parâmetros todos informados, o TBL busca qual regra, gerada a partir de um *template* instanciado, causa a maior redução de erro no estágio atual do treinamento de acordo com os valores mínimos de precisão e pontuação passados como parâmetro. Depois disso, aplica a regra encontrada aos dados de treinamento e volta a buscar e a aplicar a melhor regra do estágio atual nos dados corrigidos pela anterior. Faz isso até que a diminuição dos erros atinja um nível abaixo da pontuação ou precisão informadas, quando o aprendizado é finalizado. Finalmente tem-se uma lista de regras e a prioridade na qual devem ser aplicadas.

O processo de aprendizado de regras foi executado para 5 tipos de arquivo (chamados de gener, nume, pofs, verboc, wd, detalhados na seção 5.1), o que resultou em 5 arquivos distintos contendo regras na ordem em que devem ser aplicadas. Para todas as execuções os valores mínimos de precisão e pontuação escolhidos foram 0,5 (50%) e 5 (5 acertos a mais do que erros gerados pela regra) respectivamente e os dados de teste foram os mesmos utilizados para o treinamento.

A Tabela 5.1 mostra os totais de regras aprendidas e aplicadas, em todos os cenários (pós-edição direta e pós-edição com filtro para teste-a e teste-b), por tipo de erro e o total de *templates* dos quais essas regras foram derivadas. A partir de erros observados no corpus anotado, foram

criados manualmente 13 *templates* para nume, 7 para gener e 5 para verboc. Os demais *templates* foram criados com base em alguns exemplos fornecidos pela *u*-TBL[6] e outros exemplos mostrados em (ELMING, 2006). Pode-se notar que os valores são diferentes para as colunas "Regras Aprendidas" e "Regras Aplicadas". Isso acontece porque as ocorrências de "0" (representando palavra ou n-grama ausente) não foram tratadas pelo corretor e ocorrências de "0" (representando palavra extra) foram tratadas somente para o tipo de erro "wd". Além disso, as regras que indicam substituição para "NApp" (não aplicável) também foram desconsideradas pelo corretor.

Tabela 5.1: Totais de regras aprendidas e aplicadas por tipo de erro.

Tipo de Erro	Templates Usados	Regras Aprendidas	Regras Aplicadas
nume	18	42	30
gener	13	19	12
verboc	9	12	5
wd	6	17	5
Totais	**46**	**90**	**52**

As regras do tipo pofs não foram processadas nos experimentos descritos neste documento pois tratavam apenas de casos de palavras ausentes ou sem etiquetas morfossintáticas – que em teoria deveriam ser tratados pelo tipo wd. Duas outras regras foram aprendidas para o tipo pofs: uma delas para substituir substantivo por nome próprio e a outra para trocar determinante por verbo, o que não parecia que produziria resultados consistentes. Além disso, as regras do tipo pofs diminuíram a medida-F, calculada no momento do aprendizado das regras.

5.3 Processamento das regras

Antes de realizar o processamento das regras, as informações de nume, gener, verboc e wd são organizadas em *arrays* que têm como índice a posição do *token* no arquivo a ser traduzido. A prioridade também segue a ordem nume, gener, verboc, wd, passada como parâmetro para o EdiTA, assim os arquivos gerados pelo μ-TBL contendo regras de correção por tipo de erro foram processados nessa ordem. A cada iteração uma regra de determinado tipo é aplicada no *array* correspondente, os *arrays* corrigidos são processados usando a próxima regra, e assim por diante.

Há inúmeras variações possíveis para a aplicação das regras aprendidas. Nesse trabalho testou-se apenas uma ordem de aplicação das regras devido à limitação de tempo. Abaixo algumas ideias para experimentos futuros:

[6]Exemplos de *templates* da μ-TBL podem ser consultados em <http://www.ling.gu.se/∼lager/mutbl.html>, acesso em: 24 jan. 2014, escolhendo o *link* "Examples", e depois os *links*: "Brill Tagger", "Noun Phrase Chunker" e "Word Sense Disambiguator".

- Aplicar as regras em ordens diferentes da ordem seguida (nume, gener, verboc, wd);
- Realizar execuções considerando pofs;
- Unir as regras geradas para todos os tipos de erro em um mesmo arquivo, ordenando-as seguindo a maior combinação de pontuação e precisão;
- Ao invés de aplicar os tipos de regras a cada iteração, passando os *arrays* modificados para a verificação da próxima regra, pode-se apenas verificar a cada passagem se há alteração, "ligar" um indicador e aplicar as alterações de todos os tipos de regras apenas no final da execução.

5.4 Pós-edição dos erros de TA

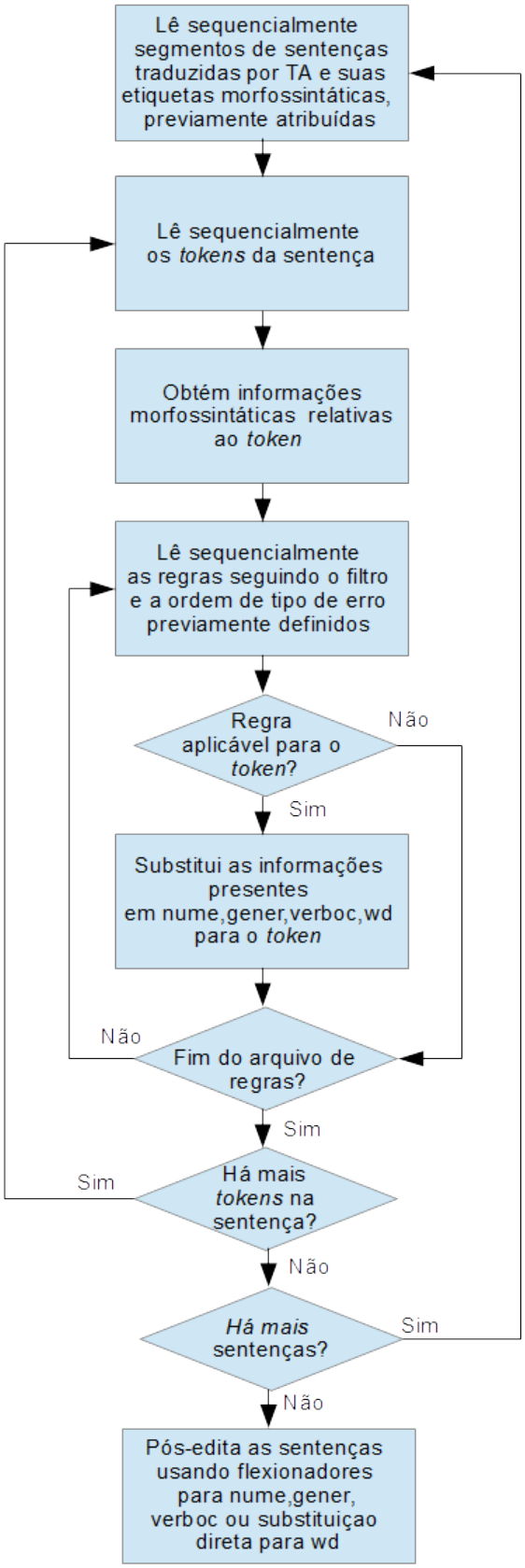
Duas formas de pós-edição foram testadas nos experimentos. Na primeira delas, chamada aqui de pós-edição direta, o identificador de erros não é usado. Nessa forma de pós-edição cada *token* da sentença tem a informação de contexto[7] verificada pelas regras a fim de decidir se será ou não pós-editado. O fluxo de processamento neste tipo de pós-edição está representado na Figura 5.6.

Para o segundo tipo de pós-edição, chamada aqui de pós-edição com filtro, três parâmetros devem ser selecionados antes do processamento:

- Tamanho do segmento ou janela: refere-se ao número máximo de *tokens* que será considerado no momento da verificação da informação de contexto pelas regras. O tamanho pode ser de 5, 7 ou 11 *tokens*. A verificação das regras é feita janela a janela, ao invés de sentença a sentença;
- Modelo treinado (por NaiveBayes, Árvores de Decisão ou SVM) aplicado para a identificação de erros;
- Formas de verificação de erro:
  - verificação geral – checa se a janela é incorreta antes de pós-editá-la ou;
  - verificação por categoria de erro – filtra por categoria de erro (A, B, C ou D) definidas na seção 3.2. As subcategorias não são consideradas aqui por não terem apresentado desempenho considerado satisfatório na tarefa de identificação automática, como pode ser visto na seção 4.3.3.

[7] A informação de contexto refere-se ao próprio *token*, e seus dados morfossintáticos, e aos *tokens* antes ou depois dele, e seus dados morfossintáticos.

Figura 5.6: Pós-edição direta, ou seja, sem passar pelo identificador de erros.



Quando a forma de verificação de erro escolhida for a "verificação por categoria de erro" a validade das regras também passa por filtro. Assim, por exemplo, as regras do tipo nume,

gener e verboc só são testadas se a categoria de erro apontada pelo identificador for da categoria A (erros morfossintáticos). Já a regra do tipo wd só é válida se a categoria identificada for a B (erros lexicais), a C (n-grama) ou a D (ordem errada). Já as regras de pofs (não usadas nos experimentos desse projeto), só são empregadas se a categoria de erro for B. A Tabela 5.2 traz as regras aplicáveis por tipo de erro. Os passos seguidos nessa forma de pós-edição podem ser vistos na Figura 5.7.

Tabela 5.2: Regras válidas para cada tipo de erro determinado pelo identificador de erros.

TIPOS DE ERRO					
R E G R A S		**A**	**B**	**C**	**D**
	gener	X			
	nume	X			
	verboc	X			
	wd		X	X	X

A Figura 5.7 mostra o fluxo da pós-edição com filtro, na qual a correção só é realizada nas sentenças automaticamente identificadas como tendo algum erro. Nas Figuras 5.6 e 5.7, o etiquetador morfossintático usado para gerar as etiquetas (citadas no primeiro passo) e os flexionadores (usados no último passo) são do Apertium (ARMENTANO-OLLER et al., 2006) e funcionam com base nos dados linguísticos originais enriquecidos no projeto ReTraTos conforme descrito em (CASELI, 2007).

Tanto na pós-edição direta como na pós-edição com filtro, após a verificação da aplicabilidade das regras, as sentenças são pós-editadas usando flexionadores de número, gênero e verbo do Apertium quando há alterações nos *arrays* dos tipos nume, gener e verboc para o *token*. Já para modificações presentes no *array* do tipo wd, há uma substituição direta do *token* que deve sofrer pós-edição pelo novo *token* contido na posição correspondente no *array* wd.

5.5 Tradutor automático TAEIP (*baseline*)

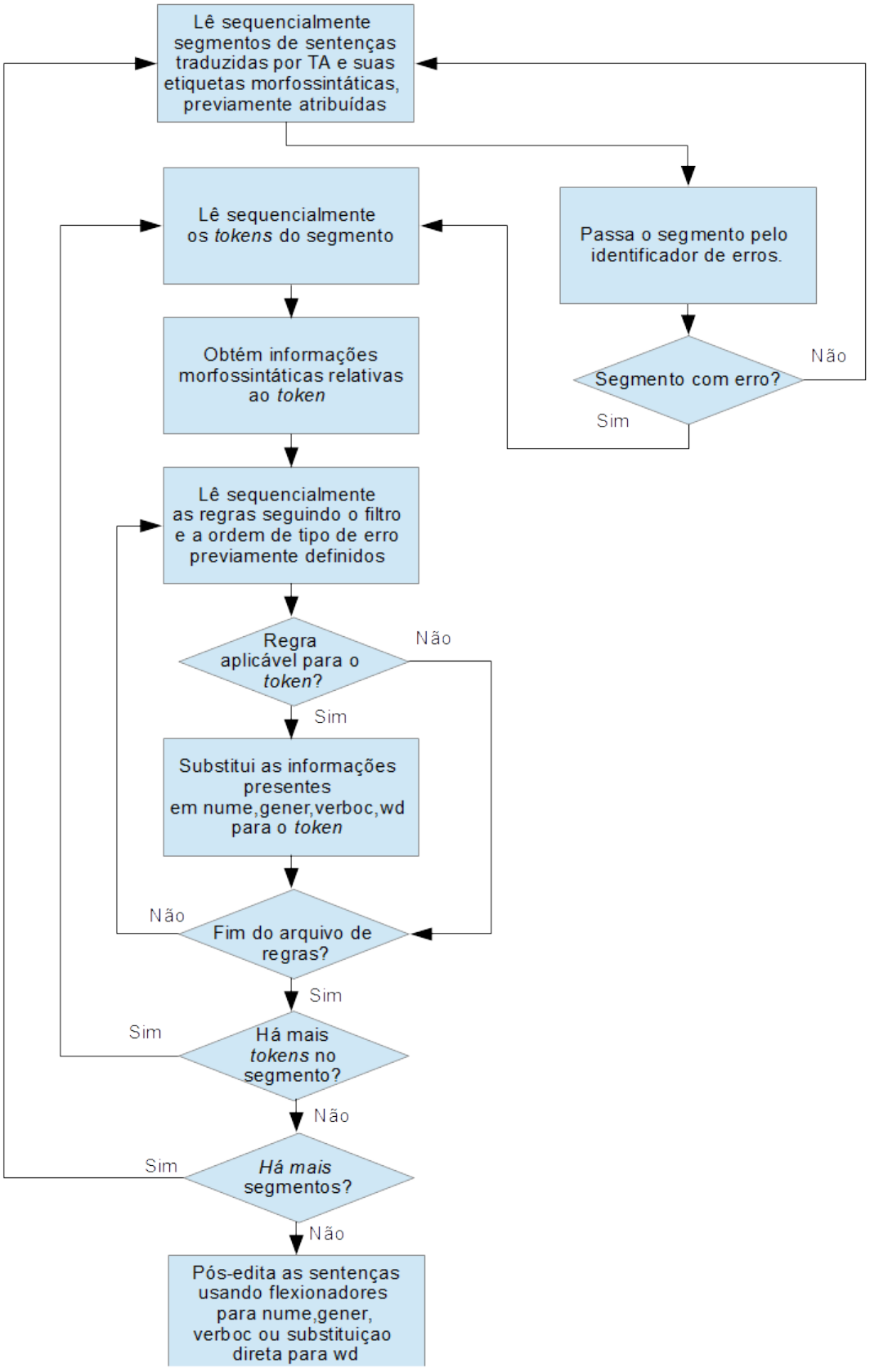
O TAEIP foi o tradutor automático selecionado como *baseline* para esta pesquisa. Assim, a saída do TAEIP para os corpora de teste é fornecida como entrada para o APE implementado nesta pesquisa e, em seguida, a saída do TAEIP sem pós-edição e a versão pós-editada são avaliadas e comparadas.

O TAEIP foi treinado com o corpus FAPESP, composto por textos da revista Pesquisa FAPESP[8] (AZIZ; SPECIA, 2011)[9]. O modelo de língua do TAEIP foi treinado com o corpus monolíngue contendo 247.625 sentenças em português, enquanto o modelo de tradução foi treinado

[8]Disponível em: <http://revistapesquisa.fapesp.br/>. Acesso em: 16 out. 2012.

[9]Disponível em: <http://pers-www.wlv.ac.uk/∼in1676/resources/fapesp/index.html>. Acesso em: 16 out. 2012.

Figura 5.7: Pós-edição com filtro. Usa o identificador de erros, ou seja, somente segmentos (janelas) com erros são passíveis de pós-edição.



usando o corpus paralelo português-inglês que possui 160.975 pares de sentenças. Além destes, também utilizou-se o corpus de otimização (*tuning*) proposto por Aziz e Specia (2011) e composto por 1.375 sentenças. O *toolkit* de SMT Moses[10] (KOEHN et al., 2007) foi executado usando os seguintes parâmetros de configuração para treinamento do TAEIP[11]:

- **Preparação do corpus**: O tamanho máximo da sentença foi limitado a 80 *tokens* para diminuir a carga combinatória na geração dos modelos, ou seja, sentenças maiores do que isso são ignoradas no processo de geração. Também utilizou-se *truecaser* como opção de pré-processamento para considerar a capitalização original dos arquivos.

- **Treinamento do modelo de língua**: A ferramenta usada para geração do modelo de língua foi o SRILM[12] (STOLCKE, 2002) com especificação dos seguintes parâmetros: *-interpolate -kndiscount -unk* e ordem do modelo de língua igual a no máximo 5 *tokens*.

- **Treinamento do modelo de tradução**: O método de simetrização (combinação) dos alinhamentos fonte-alvo e alvo-fonte do alinhador lexical usado no treinamento do modelo de tradução, o GIZA++[13] (OCH; NEY, 2003), foi o *grow-diag-final-e* e o método de reordenação utilizado, o *wbe-msd-bidirectional-fe*. O tamanho máximo do n-grama considerado na geração do modelo de tradução foi 7 *tokens*. Os parâmetros aplicados para o GIZA++ foram:

  - 5 iterações Modelo1 IBM
  - 3 iterações Modelo3 IBM
  - 3 iterações Modelo4 IBM
  - 5 iterações do *Hidden Markov Models* (HMM)

- **Otimização (*Tuning*)**: Por fim, a otimização dos modelos foi realizada com base na lista de 100 melhores (*n-best list*) traduções.

Os modelos estatísticos derivados desse treinamento constituem o TAEIP e podem ser obtidos no Portal de TA, PorTAl.[14]

[10]Disponível em: <http://www.statmt.org/moses/>. Acesso em: 16 out. 2012.

[11]Para obter detalhes sobre o funcionamento das ferramentas citadas ou o significado dos parâmetros utilizados sugere-se consultar as páginas oficiais das ferramentas.

[12]Disponível em: <http://www.speech.sri.com/projects/srilm/download.html>. Acesso em: 20 jan. 2014.

[13]Disponível em: <http://www.statmt.org/moses/giza/GIZA++.html>. Acesso em: 20 jan. 2014.

[14]Disponível em: <http://www.lalic.dc.ufscar.br/portal/>. Acesso em: 16 out. 2012.

5.6 Experimentos para correção automática de erros

Uma vez que o TAEIP foi o tradutor automático usado como *baseline* nestes experimentos, a principal hipótese de trabalho aqui perseguida é a de que haja melhora na qualidade das traduções quando o EdiTA é aplicado para pós-editar a saída do TAEIP. Outra hipótese investigada neste trabalho é a de que a pós-edição feita em duas etapas – identificação de erros seguida pela correção desses erros – evita que correções desnecessárias, que podem dar origem a novos erros, sejam efetuadas pelo corretor.

Os testes realizados com pós-edição, tanto para o conjunto de treinamento quanto para o conjunto de teste, foram os de pós-edição direta (seção 5.6.1) e de pós-edição com filtro usando o algoritmo árvore de decisão e janelas de tamanhos 5, 7 e 11 (seção 5.6.2). Optou-se por usar os modelos treinados com árvores de decisão para a identificação de erros pois este algoritmo obteve o melhor desempenho geral nos experimentos apresentados na seção 4.3. Vale ressaltar que embora não seja usual avaliar o sistema no mesmo corpus no qual foi treinado (corpus de treinamento), adotou-se essa estratégia para: (i) verificar qual seria o patamar superior de precisão e cobertura ao qual o EdiTA poderia chegar e (ii) verificar se o método de aprendizado utilizado tem tendência a *overfitting*.

A avaliação foi realizada com os corpora propostos por Aziz e Specia (2011), contendo 1.314 (teste-a, corpus de treinamento desta pesquisa) e 1.447 (teste-b, corpus de teste) sentenças, para as medidas de avaliação BLEU (PAPINENI et al., 2002) e NIST (DODDINGTON, 2002) (veja seção 2.1.2).

Além dessas medidas usadas para a avaliação geral da TA, sempre que houver referência à precisão ou à cobertura de uma regra deve-se atentar que o cálculo é feito com base nas equações 5.3 e 5.4, respectivamente. A precisão é calculada como a soma das vezes em que a regra foi aplicada resultando em melhora no BLEU (melhoraBLEU(R)) ou em que o BLEU ficou igual (igualBLEU(R)) dividida pelo total de vezes em que a regra foi aplicada. Já a cobertura, disponível apenas para o conjunto de treinamento (onde as ocorrências de erros foram manualmente anotadas) é calculada como a soma das vezes em que a regra foi aplicada resultando em melhora no BLEU (melhoraBLEU(R)) ou em que o BLEU ficou igual (igualBLEU(R)) dividida pelo total de erros anotados para o tipo de regra sendo aplicada (erros(R)).

$$precisao(R) = \frac{|melhoraBLEU(R)| + |igualBLEU(R)|}{|melhoraBLEU(R)| + |pioraBLEU(R)| + |igualBLEU(R)|} \tag{5.3}$$

$$cobertura(R) = \frac{|melhoraBLEU(R)| + |igualBLEU(R)|}{|erros(R)|} \tag{5.4}$$

Como as medidas automáticas BLEU e NIST baseiam-se no casamento de n-gramas entre a tradução automática e a referência, seus valores podem não refletir as melhoras ocasionadas pela alteração de um único *token* com a aplicação de uma regra de correção. Essa limitação se estende para precisão (equação 5.3) e cobertura (equação 5.4) das regras, definidas neste trabalho com base nos valores de BLEU. Assim, para complementar a avaliação com base em medidas automáticas, também foram realizadas análises manuais em alguns casos como relatado nas seções a seguir.

5.6.1 Experimentos usando pós-edição direta

Nos experimentos com pós-edição direta (sem a etapa prévia de identificação de erros), notou-se que os valores de BLEU e NIST permaneceram praticamente inalterados em relação aos valores obtidos para a tradução sem pós-edição (saída da TA), tanto no corpus de treinamento como no de teste. O resultado pode ser visto na Tabela 5.3.

Tabela 5.3: Valores de BLEU e NIST dos conjuntos de treinamento e de teste para a saída da TA (sem pós-edição) e com pós-edição direta.

Corpus	Avaliação	Saída da TA	Pós-edição direta
Teste-a (treinamento)	**BLEU**	60,02	60,10
	NIST	10,96	10,97
Teste-b (teste)	**BLEU**	49,59	49,54
	NIST	9,81	9,81

Analisando-se a precisão e a cobertura das regras aplicadas na pós-edição direta têm-se os valores apresentados na Tabela 5.4 considerando-se a aplicação de cada conjunto de regras separadamente e de todas as regras na ordem nume, gener, verboc e wd. Como esperado, a precisão no conjunto de treinamento ficou maior que a do conjunto de teste. Um valor baixo de cobertura também era esperado, pois o cálculo é realizado sobre todos os erros anotados para cada tipo de regra (301 para nume, 271 para gener, 209 para verboc e 988 para wd).

Tabela 5.4: Valores de precisão (%) e cobertura (%) na aplicação de regras para a saída da TA com pós-edição direta considerando-se a aplicação de cada conjunto de regras separadamente e de todas.

Corpus	Avaliação	nume	gener	verboc	wd	todas
Teste-a (treinamento)	**Precisão**	90,41	88,89	94,73	90,91	88,23
	Cobertura	21,93	29,52	8,61	1,01	8,25
Teste-b (teste)	**Precisão**	53,37	83,33	41,18	20,00	65,79

Fazendo uma análise tipo a tipo, vê-se que as regras do tipo gener, quando aplicadas isoladamente, foram as que obtiveram a maior cobertura (29,5%), além disso elas tiveram um

desempenho no conjunto de teste (83%) similar ao desempenho no conjunto de treinamento (89%), indicando que foi realizado um bom aprendizado e que essas regras são as mais generalizáveis. O oposto ocorre com as regras do tipo wd, que quando aplicadas em separado, demonstram-se muito eficientes no conjunto de treinamento (91% de precisão) e apresentam uma baixa precisão (20%) no conjunto de teste, indicando um *overfitting.*

As Tabelas 5.5 e 5.6 listam as 10 regras mais aplicadas nos experimentos com pós-edição direta considerando-se os corpora teste-a e teste-b, respectivamente. Cada regra é acompanhada do número de vezes em que foi aplicada (Aplicações) e a precisão individual da regra calculada de acordo com a equação 5.3.

Tabela 5.5: As 10 regras mais aplicadas, em ordem decrescente por número de aplicações, para a pós-edição direta no corpus teste-a , acompanhadas do número de aplicações e precisão (%).

Regra	Aplicações	Precisão
gener:m>f <- gener:0@[1]	49	83,67
gener:f>m <- gener:m@[-1] & gener:m@[1]	14	85,71
nume:pl>sg <- pofs:n@[-2] & pofs:adv@[-1] & nume:sg@[-1,-2,-3] o	12	75,00
nume:sg>pl <- nume:sg@[0] & verboc:'ind.sg.p1.futpret'@[0] & nume:pl@[-1,-2,-3]	9	88,89
verboc:ind.sg.p1.futpret>ind.pl.p3.futpret <- nume:pl@[-1,-2,-3]	9	88,89
gener:m>f <- pofs:num@[1] & pofs:n@[2] & gener:f@[1,2,3]	6	83,33
gener:f>m <- pofs:preadv@[-1] & gener:NApp@[1,2,3]	5	60,00
gener:mf>m <- pofs:0@[1] & pofs:0@[2] & gener:NApp@[1,2,3]	4	100,00
nume:sg>pl <- nume:sg@[0] & nume:pl@[1,2,3,4] & wd:circuitos@[1,2,3]	4	100,00

Tabela 5.6: As 10 regras mais aplicadas, em ordem decrescente por número de aplicações, para a pós-edição direta no corpus teste-b, acompanhas do número de aplicações e precisão (%).

Regra	Aplicações	Precisão
gener:f>m <- gener:m@[-1] & gener:m@[1]	12	83,33
gener:NApp>m <- gener:NApp@[0] & gener:m@[-2,-3,-4] & pofs:pr+det@[-1] & pofs:n@[1]	10	100,00
nume:pl>sg <- pofs:n@[-2] & pofs:adv@[-1] & nume:sg@[-1,-2,-3]	9	66,67
nume:sg>pl <- nume:sg@[0] & verboc:ind.sg.p1.futpret@[0] & nume:pl@[-1,-2,-3]	9	33,33
verboc:ind.sg.p1.futpret>ind.pl.p3.futpret <- nume:pl@[-1,-2,-3]	9	33,33
gener:m>f <- pofs:num@[1] & pofs:n@[2] & gener:f@[1,2,3]	5	80,00
verboc:pp.pl.NApp.NApp>pp.sg.NApp.NApp <- pofs:pr@[1] & pofs:NC@[2] & verboc:NApp@[1,2,3]	4	100,00
wd:para>0 <- wd:entender@[1]	3	33,33
nume:NApp>pl <- pofs:n@[-2] & pofs:cnjsub@[-1] & nume:NApp@[-1,-2,-3]	2	100,00
nume:pl>sg <- pofs:v@[-2] & pofs:preadv@[-1] & nume:NApp@[1,2,3]	2	50,00

Além da avaliação de todo o arquivo pós-editado, também verificou-se os valores de BLEU e NIST sentença a sentença para os conjuntos de treinamento e teste. A essa análise dos valores das métricas para cada sentença somou-se uma checagem manual a nível sentencial.[15] Os

[15]Na verificação manual considerou-se que houve melhora se pelo menos um dos problemas da palavra foi

valores dessa verificação para pós-edição direta quando todas as regras foram aplicadas conjuntamente são mostrados na Tabela 5.7.

Tabela 5.7: Quantidade de sentenças que melhoraram ou pioraram de acordo com os valores de BLEU e NIST, e verificação manual para a pós-edição direta quando todas as regras foram aplicadas conjuntamente.

Corpus	Avaliação	Melhora	Piora
Teste-a (treinamento)	**BLEU**	37	18
	NIST	38	19
	Manual	76	29
Teste-b (teste)	**BLEU**	5	18
	NIST	5	18
	Manual	20	31

Como é possível notar pelos valores desta tabela, o avaliador humano detectou um número maior de alterações (tanto melhora quanto piora) do que o que foi detectado com base nas medidas automáticas BLEU e NIST, isso porque o avaliador analisou especificamente as alterações realizadas pelas regras o que as medidas automáticas não são capazes de fazer. Com base na análise manual também vale notar que, das alterações realizadas pelas regras, 72% foram para melhor no corpus de treinamento e apenas 39%, no de teste.

A fim de demonstrar o resultado da aplicação das regras, foram separados alguns exemplos. As Figuras 5.8, 5.9 e 5.10 trazem trechos de sentenças do corpus de teste (teste-b) pós-editados pelo EdiTA aplicando pós-edição direta. A palavra pós-editada aparece em destaque para “Ape”, assim como suas correspondentes em Src, Ref e Sys. Nos dois primeiros exemplos a alteração foi executada de forma correta. No exemplo representado pela Figura 5.8 houve um aumento de BLEU, já no exemplo da Figura 5.9, apesar da pós-edição ter sido aplicada corretamente, o BLEU permaneceu inalterado devido ao uso de um verbo diferente pelo TAEIP quando comparado com o utilizado na referência. No terceiro exemplo (Figura 5.10) a pós-edição gerou uma diminuição no BLEU: o etiquetador do Apertium induziu o corretor ao erro, já que a palavra “bateria” foi etiquetada como sendo um verbo (lemma=“bater” pos=“v” form=“ind” number=“sg” person=“p1” time=“futpret”), quando na verdade trata-se de um substantivo.

5.6.2 Experimentos usando pós-edição com filtro

Nos experimentos usando filtro, tanto para incorretos como por categoria de erro, novamente os valores de BLEU e NIST permaneceram praticamente inalterados em relação aos valores obtidos para a tradução sem pós-edição (saída da TA), tanto no corpus de treinamento como no de teste. As Tabelas 5.8 e 5.9 trazem os resultados detalhados.

corrigido (concordância de número, gênero, etc.) ainda que a palavra tenha mais de um erro. Quando na dúvida, ficou indicado pelo avaliador que não houve nem melhora nem piora.

Figura 5.8: Erro de concordância em gênero pós-editado corretamente. O BLEU passou de 39,73 para 41,96 para a sentença.

Regra	gener:m>f <- pofs:num@[1] & pofs:n@[2] & gener:f@[1,2,3]
Src	*Of the 150 Brazilian Indian languages , at least 21 % are seriously threatened...*
Ref	Das 150 línguas indígenas , pelo menos 21 % delas estão seriamente ameaçadas...
Sys	Dos 150 línguas indígenas brasileiros , pelo menos 21 % estão seriamente ameaçados...
Ape	Das 150 línguas indígenas brasileiros , pelo menos 21 % estão seriamente ameaçados...

Figura 5.9: Erro de concordância em número pós-editado corretamente. O BLEU permaneceu inalterado em 54,97.

Regra	verboc:inf.NApp.NApp.NApp>ind.pl.p3.pres<- pofs:n@[-2] & pofs:cnjsub@[-1] & verboc:NApp@[1,2,3]
Src	*... articles and books that explain the results of original research ...*
Ref	... artigos e livros que exponham resultados originais de pesquisa ...
Sys	... artigos e livros que explicar os resultados de pesquisa original ...
Ape	... artigos e livros que explicam os resultados de pesquisa original ...

Figura 5.10: Pós-edição executada incorretamente. O BLEU passou de 75,71 para 68,14 para a sentença.

Regra	nume:sg>pl <- nume:sg@[0] & verboc:ind.sg.p1.futpret@[0] & nume:pl@[-1,-2,-3]
Src	*The energy expended by our battery is 60 watt @-@ hours ...*
Ref	A energia despendida pela nossa bateria é de 60 watts @-@ hora ...
Sys	A energia empreendidos pela nossa bateria é de 60 watts @-@ horas ...
Ape	A energia empreendidos pela nossa bateriam é de 60 watts @-@ horas ...

Tabela 5.8: Valores de BLEU e NIST dos conjuntos de treinamento e de teste para a saída da TA (sem pós-edição), com pós-edição direta e com pós-edição aplicando filtro para sentenças incorretas com diferentes tamanhos de janela.

Corpus	Avaliação	Saída da TA	Pós-edição direta	TW-5	TW-7	TW-11
Teste-a (treinamento)	**BLEU**	60,02	60,10	60,07	60,10	60,08
	NIST	10,96	10,97	10,97	10,97	10,97
Teste-b (teste)	**BLEU**	49,59	49,54	49,59	49,59	49,58
	NIST	9,81	9,81	9,81	9,81	9,81

A precisão e a cobertura do total das regras aplicadas podem ser encontradas nas Tabelas 5.10 e 5.11 para o filtro por incorreto e por categoria de erro, respectivamente. Como es-

Tabela 5.9: Valores de BLEU e NIST dos conjuntos de treinamento e de teste para a saída da TA (sem pós-edição), com pós-edição direta e com pós-edição aplicando filtro para categorias de erro com diferentes tamanhos de janela.

Corpus	Avaliação	Saída da TA	Pós-edição direta	TW-5	TW-7	TW-11
Teste-a (treinamento)	**BLEU**	60,02	60,10	60,07	60,09	60,09
	NIST	10,96	10,97	10,97	10,97	10,97
Teste-b (testes)	**BLEU**	49,59	49,54	49,59	49,59	49,60
	NIST	9,81	9,81	9,81	9,81	9,81

perado, a precisão usando pós-edição com filtro é maior do que usando pós-edição direta e sua cobertura, menor (veja Tabela 5.4).

Tabela 5.10: Valores de precisão (%) e cobertura (%) aplicando filtro para sentenças incorretas antes da pós-edição considerando-se a aplicação de cada conjunto de regras separadamente e de todas.

Janela	Corpus	Avaliação	nume	gener	verboc	wd	todas
5	Teste-a (treinamento)	**Precisão**	100,00	100,00	N/A	100,00	100,00
		Cobertura	4,32	3,32	N/A	0,40	1,38
	Teste-b (testes)	**Precisão**	100,00	83,33	N/A	25,00	69,23
7	Teste-a (treinamento)	**Precisão**	100,00	100,00	N/A	100,00	100,00
		Cobertura	4,65	5,17	N/A	0,40	1,71
	Teste-b (testes)	**Precisão**	100,00	75,00	N/A	20,00	64,71
11	Teste-a (treinamento)	**Precisão**	93,33	92,31	N/A	100,00	93,33
		Cobertura	4,65	4,43	N/A	0,30	1,54
	Teste-b (testes)	**Precisão**	83,33	72,73	N/A	20,00	63,64

Tabela 5.11: Valores de precisão (%) e cobertura (%) aplicando filtro por categoria de erro antes da pós-edição considerando-se a aplicação de cada conjunto de regras separadamente e de todas.

Janela	Corpus	Avaliação	nume	gener	verboc	wd	todas
5	Teste-a (treinamento)	**Precisão**	100,00	100,00	N/A	100,00	100,00
		Cobertura	3,32	2,21	N/A	0,40	1,10
	Teste-b (testes)	**Precisão**	N/A	50,00	N/A	N/A	50,00
7	Teste-a (treinamento)	**Precisão**	100,00	100,00	N/A	100,00	100,00
		Cobertura	3,65	4,06	N/A	0,40	1,43
	Teste-b (testes)	**Precisão**	100,00	80,00	N/A	N/A	83,33
11	Teste-a (treinamento)	**Precisão**	91,67	100,00	N/A	100,00	96,43
		Cobertura	3,65	4,43	N/A	0,40	1,49
	Teste-b (testes)	**Precisão**	100,00	76,92	N/A	N/A	81,25

Ambas as tabelas trazem também valores de precisão e cobertura considerando a aplicação de cada conjunto de regras separadamente e de todas as regras na ordem nume, gener, verboc e wd (coluna todas).

Assim como nos testes com pós-edição direta (seção 5.6.1), para a pós-edição com filtro, também foi realizada uma análise regra a regra listando o número de aplicações e a precisão

(equação 5.3) de cada uma delas. As Tabelas 5.12 e 5.13, trazem as regras aplicadas usando o filtro de incorretos, para os corpora teste-a e teste-b, respectivamente. Para os filtros por categoria de erros, as regras aplicadas estão nas Tabelas 5.14 e 5.15. Os valores listados, tanto para filtro de incorretos quanto para o filtro por categoria de erros, referem-se à execução usando a janela de tamanho 7, pois foi a que apresentou o melhor desempenho na verificação manual.

Tabela 5.12: Regras aplicadas para a pós-edição com filtro de incorretos e janela de tamanho 7, em ordem decrescente por número de aplicações, no corpus teste-a, acompanhas do número de aplicações e precisão (%).

Regra	Aplicações	Precisão
gener:f>m <- gener:m@[-1] & gener:m@[1]	10	100,00
nume:pl>sg <- pofs:n@[-2] & pofs:adv@[-1] & nume:sg@[-1,-2,-3]	8	100,00
gener:m>f <- pofs:num@[1] & pofs:n@[2] & gener:f@[1,2,3]	4	100,00
wd:por>de <- wd:quente@[-1]	3	100,00
nume:sg>pl <- nume:sg@[0] & nume:pl@[1,2,3,4] & wd:sacos@[1,2,3]	2	100,00
nume:sg>pl <- nume:sg@[0] & nume:pl@[1,2,3,4] & wd:usos@[1,2,3]	2	100,00
nume:sg>pl <- nume:sg@[0] & nume:sg@[1,2,3,4] & wd:trellises@[1,2,3]	1	100,00
wd:para>0 <- wd:entender@[1]	1	100,00

Tabela 5.13: Regras aplicadas, em ordem decrescente por número de aplicações, para a pós-edição com filtro de incorretos e janela de tamanho 7 no corpus teste-b, acompanhas do número de aplicações e precisão (%).

Regra	Aplicações	Precisão
gener:f>m <- gener:m@[-1] & gener:m@[1]	6	83,33
nume:pl>sg <- pofs:n@[-2] & pofs:adv@[-1] & nume:sg@[-1,-2,-3]	4	100,00
wd:para>0 <- wd:entender@[1]	4	25,00
gener:m>f <- pofs:num@[1] & pofs:n@[2] & gener:f@[1,2,3]	2	50,00
wd:de>que <- wd:a@[1]	1	0,00

Tabela 5.14: Regras aplicadas para a pós-edição com filtro por categoria de erro, em ordem decrescente por número de aplicações, no corpus teste-a, acompanhas do número de aplicações e precisão (%).

Regra	Aplicações	Precisão
nume:pl>sg <- pofs:n@[-2] & pofs:adv@[-1] & nume:sg@[-1,-2,-3]	9	100,00
gener:f>m <- gener:m@[-1] & gener:m@[1]	8	100,00
gener:m>f <- pofs:num@[1] & pofs:n@[2] & gener:f@[1,2,3]	3	100,00
wd:por>de <- wd:quente@[-1]	3	100,00
nume:sg>pl <- nume:sg@[0] & nume:pl@[1,2,3,4] & wd:usos@[1,2,3]	2	100,00
wd:para>0 <- wd:entender@[1]	1	100,00

Além da avaliação de todo o arquivo pós-editado, também verificou-se os valores de BLEU e NIST sentença a sentença para os conjuntos de treinamento e teste. A essa análise dos valores das métricas para cada sentença somou-se uma checagem manual no conjunto de teste a nível

Tabela 5.15: Regras aplicadas para a pós-edição com filtro por categoria de erro, em ordem decrescente por número de aplicações, no corpus teste-b, acompanhas do número de aplicações e precisão (%).

Regra	Aplicações	Precisão
gener:m>f <- pofs:num@[1] & pofs:n@[2] & gener:f@[1,2,3]	3	66,67
gener:f>m <- gener:m@[-1] & gener:m@[1]	2	100,00
nume:pl>sg <- pofs:n@[-2] & pofs:adv@[-1] & nume:sg@[-1,-2,-3]	1	100,00

sentencial. O resultado para a pós-edição usando filtro de incorretos está na Tabela 5.16 e os valores aplicando filtro por categoria de erro estão na Tabela 5.17.

Tabela 5.16: Quantidade de sentenças que melhoraram ou pioraram – valores de BLEU e NIST do conjunto de treinamento e BLEU, NIST e verificação manual do conjunto de teste para a pós-edição com filtro para incorretos.

		Janela=5		Janela=7		Janela=11	
Corpus	**Avaliação**	**Melhora**	**Piora**	**Melhora**	**Piora**	**Melhora**	**Piora**
Teste-a (treinamento)	**BLEU**	12	0	16	0	15	2
	NIST	12	0	16	0	15	2
Teste-b (teste)	**BLEU**	3	4	3	5	3	7
	NIST	3	4	3	5	3	7
	Manual	7	4	9	4	9	7

Tabela 5.17: Quantidade de sentenças que melhoraram ou pioraram – valores de BLEU e NIST do conjunto de treinamento e BLEU, NIST e verificação manual do conjunto de teste para a pós-edição com filtro por categoria de erro.

		Janela=5		Janela=7		Janela=11	
Corpus	**Avaliação**	**Melhora**	**Piora**	**Melhora**	**Piora**	**Melhora**	**Piora**
Teste-a (treinamento)	**BLEU**	11	0	14	0	15	1
	NIST	11	0	14	0	15	1
Teste-b (teste)	**BLEU**	0	1	1	1	2	3
	NIST	0	1	1	1	2	3
	Manual	2	0	6	0	9	2

Novamente, como é possível notar pelos valores das tabelas 5.16 e 5.17, o avaliador humano detectou um número maior de alterações (tanto melhora quanto piora) do que o que foi detectado com base nas medidas automáticas BLEU e NIST. Com base na análise manual também vale notar que, para o melhor tamanho de janela (7), das alterações realizadas pelas regras, 69% foram para melhor no corpus de teste com filtro de incorretos e 100% no filtro de categoria, ambos valores maiores do que os 39% obtidos na pós-edição direta.

Para demonstrar o resultado da aplicação das regras, foram separados alguns exemplos representados pelas Figuras 5.11, 5.12, 5.13 e 5.14, que trazem trechos de sentenças do corpus de teste (teste-b) pós-editados pelo EdiTA usando pós-edição com filtro. A palavra pós-editada

aparece em destaque para “Ape”, assim como suas correspondentes em Src, Ref e Sys. Nos dois primeiros exemplos a correção foi executada de forma correta. No primeiro exemplo (Figura 5.11) houve um aumento de BLEU, já no segundo exemplo (Figura 5.12), apesar da pós-edição ter sido aplicada corretamente, o BLEU permaneceu inalterado porque apenas o erro de concordância em gênero foi corrigido e o erro de concordância em número não.

Dois exemplos de pós-edições incorretas do EdiTA usando filtro podem ser vistos nas Figuras 5.13 e 5.14. No segundo deles (Figura 5.14) o erro ocorreu devido a uma diferença na forma como o Apertium e o EdiTA processam os *tokens*. Isso se dá porque os dados para treinar o corretor, e executar a correção, são processados *token* a *token*, enquanto o Apertium une *tokens* em alguns casos da etiquetação. A união de dois *tokens* pelo Apertium está prevista no EdiTA. Entretanto, na sentença mostrada na Figura 5.14, houve a união de três *tokens*: “no”, “final” e “do” (no_final_de+o), gerando um erro de alinhamento entre as duas ferramentas.

Figura 5.11: Erro de concordância em número pós-editado corretamente. O BLEU passou de 55,06 para 62,03.

Regra	nume:pl>sg <- pofs:n@[-2] & pofs:adv@[-1] & nume:sg@[-1,-2,-3]
Src	*... it can also be used in vessels with a more complex anatomy , ” wrote the experts ...*
Ref	... pode permitir , ainda , sua utilização em vasos com anatomia mais complexa ” , escreveram os especialistas ...
Sys	... pode ser usado também em vasos com anatomia mais complexas ” , escreveram os especialistas ...
Ape	... pode ser usado também em vasos com anatomia mais complexa ...” , escreveram os especialistas ...

Figura 5.12: Erro de concordância em gênero pós-editado corretamente. O BLEU permaneceu inalterado em 54,67.

Regra	gener:f>m <- gener:m@[-1] & gener:m@[1]
Src	*... the formation of the mountains – and , therefore , the closing off of the Clymene ocean ...*
Ref	... a formação das montanhas - e , portanto , o fechamento do oceano Clymene ...
Sys	... a formação das montanhas – e , portanto , o fechamento das oceano Clymene ...
Ape	... a formação das montanhas – e , portanto , o fechamento dos oceano Clymene ...

Comparando-se os valores das Tabelas 5.7 e 5.16 é possível notar que apesar da pós-edição ser mais restrita quando passada pelo identificador (pós-edição com filtro), a porcentagem de sentenças pós-editadas que apresentaram melhora aumentou significativamente quando comparados aos valores obtidos na pós-edição direta (Tabela 5.7). Considerando o BLEU e o corpus

Figura 5.13: Eliminação incorretamente executada. O BLEU passou de 64,19 para 63,06 para a sentença.

Regra	wd:para>0 <- wd:entender@[1]
Src	*This is important in order to understand why certain salts cause the precipitation ...*
Ref	Esse conhecimento será importante para compreender por que determinados sais induzem a precipitação ...
Sys	Isso é importante para entender por que determinados sais causam a precipitação ...
Ape	Isso é importante entender por que determinados sais causam a precipitação ...

Figura 5.14: Pós-edição incorretamente executada. O BLEU passou de 34,41 para 33,95 para a sentença.

Regra	gener:f>m <- gener:m@[-1] & gener:m@[1]
Src	*According to him , in the late twentieth century and early twenty @-@ first century , there was a lot of success employing probabilistic methods in the study of deterministic problems .*
Ref	Segundo ele , no final do século XX e no início do XXI houve um grande sucesso no emprego de métodos probabilísticos para o estudo de problemas determinísticos .
Sys	Segundo ele , no final do século XX e início do século XXI , havia muito sucesso utilizando métodos anotação no estudo de deterministas problemas .
Ape	Segundo ele , no final do século XX e início do século XXI , havia muito sucesso utilizando anotação anotação no estudo de deterministas problemas .

teste-a, usando pós-edição direta, 67% das sentenças alteradas foram melhoradas. Já com o uso do identificador de erros esse valor passa para 100%, tanto aplicando o filtro de incorretos como por categoria de erros, quando considerada a janela de tamanho 7. Ao analisar a verificação manual, realizada para o corpus teste-b, usando pós-edição direta, pode-se notar que 39% das sentenças alteradas obtiveram melhora. Quando aplicado o identificador de erros esse valor passa para 69% usando o filtro de incorretos e 100% quando filtrado por categoria de erros, considerando para ambos os filtros a janela de tamanho 7.

5.7 Discussão dos resultados para correção automática de erros

Neste capítulo foram apresentados experimentos para correção automática da saída da TA sem (pós-edição direta) e com (pós-edição com filtro) a etapa prévia de identificação automática

de erros. Considerando-se apenas as medidas automáticas de avaliação geral da TA BLEU e NIST, não é possível concluir que a utilização do EdiTA melhora a saída do tradutor usado como *baseline*, o TAEIP. Contudo, realizando-se uma análise mais detalhada em relação à precisão e à cobertura das regras, bem como a análise manual das alterações por elas realizadas, conclui-se que o EdiTA tem um impacto positivo na saída do TAEIP.

Como é possível notar pelos valores das Tabelas 5.7, 5.16 e 5.17, na pós-edição direta o número de sentenças pós-editadas foi maior do que na pós-edição com filtro. No entanto, nem sempre a alteração realizada pelo pós-editor direto trouxe melhora na saída da TA sendo melhor, por exemplo, em apenas 39% dos casos alterados no corpus de teste (teste-b) contra 69% e 100% no pós-editor com filtro de incorretos e categorias de erros, respectivamente. Esse fato é um indício para uma das hipóteses de trabalho desta pesquisa (H2): a identificação de erros como passo prévio à correção deve ser aplicada para evitar a geração de ruídos, ou seja, a alteração desnecessária de trechos corretos.

Em relação à principal hipótese de trabalho (H1), também considerando-se os valores das tabelas citadas, é possível concluir que pode-se pós-editar automaticamente as traduções uma vez que bons resultados foram obtidos com um corpus de tamanho bastante limitado (apenas 2.147 instâncias de treinamento).

Vale notar, também, que a pequena quantidade de sentenças alteradas pela aplicação do EdiTA no corpus de teste (teste-b) explica porque os valores de BLEU e NIST permaneceram praticamente inalterados. Além disso, a pequena quantidade de sentenças pós-editadas nesse corpus em relação à quantidade de sentenças pós-editadas no corpus de treinamento (teste-a) é um indício de que o aprendiz de regras μ-TBL é altamente dependente do corpus de treinamento (um indício de *overfitting*). Esse é um problema que pode ser resolvido aumentando-se e diversificando-se o corpus de treinamento.

Onde ocorreu uma verificação manual, pôde-se notar que os avaliadores automáticos BLEU e NIST penalizaram o sistema quando comparados com a avaliação feita por um humano, tanto na pós-edição com filtro como na sem filtro. A verificação manual levou em conta que houve melhora se pelo menos um dos problemas da palavra foi corrigido (concordância em número, concordância em gênero, forma verbal, tempo verbal, etc.) ainda que a palavra tenha mais de um erro. Quando houve dúvida, ficou indicado pelo avaliador que não aconteceu nem melhora nem piora.

Outra importante constatação derivada dos experimentos foi a de que o EdiTA está sujeito a ruídos (distorções nos resultados) em todas as etapas do processo, pois envolve o uso de várias ferramentas. Alguns momentos em que tais ruídos podem ocorrer são:

1. no alinhamento entre Src e Sys, afetando a geração de *features* de treinamento do identi-

ficador de erros;

2. na etiquetação morfossintática e no alinhamento entre o corpus anotado (*token* a *token*) e as unidades lexicais resultantes do etiquetador (que pode unir vários *tokens*), ambos impactando tanto no treinamento do identificador de erros como no treinamento do corretor automático;

3. no alinhamento entre Sys e Ref, o que interfere no treinamento do corretor automático;

4. no flexionador usado para conjugação verbal, de gênero e de número, resultando em ruídos na fase final da pós-edição;

5. na ferramenta usada para o aprendizado de regras de correção automática que tem limitações como o aprendizado de regras de correção apenas para um tipo de erro de cada vez;

6. no código interno do próprio EdiTA, como a incompatibilidade entre trechos identificados com erros (na primeira etapa do pós-editor com filtro) e o escopo de aplicação das regras (na segunda etapa do pós-editor com filtro). Esse pode ser o motivo da não aplicação de regras de correção do tipo verboc no pós-editor com filtro;

7. no processo manual de anotação de erros no corpus usado para treinamento.

A partir de tudo o que foi apresentado neste capítulo, apesar do EdiTA não ter causado mudanças significativas nos valores calculados pelas medidas BLEU e NIST, em comparação com os obtidos pelo TAEIP, esta pesquisa provou as hipóteses que se propôs a investigar pois, conforme verificado na análise detalhada de precisão e cobertura das regras e na análise manual: (H1) a pós-edição automática melhora a qualidade da TA e (H2) a identificação de erros realizada como passo prévio à correção evita que alterações desnecessárias sejam realizadas. Sobre a primeira hipótese, H1, quando foi executada a pós-edição direta (PED), ou seja, sem filtros, a precisão na aplicação das regras no corpus de teste atingiu cerca de 66%, aplicando-se todas as regras. Quando testada a segunda hipótese, H2, também utilizando todas as regras, a cobertura ficou mais baixa (entre 1 e 1,7% analisando os dois tipos de filtro e as três janelas) em comparação com a cobertura obtida na PED (8,25%). No entanto, a maior precisão atigida no conjunto de teste (83%, filtro por categorias de erro, janela de tamanho 7) ficou bem acima dos 66% obtidos sem o uso de filtros. Ou seja, em H2 há menos alterações, mas tende-se a cometer menos erros, pois as pós-edições ocorrem apenas em trechos que realmente precisam ser alterados.

Embora os resultados obtidos nesta pesquisa apontem para uma direção promissora, experimentos adicionais precisam ser executados no futuro com o intuito de aumentar a qualidade da pós-edição. Algumas linhas que poderão ser investigadas são:

- Aumentar o tamanho do corpus de treinamento;
- Otimizar o treinamento utilizando um corpus de validação diferente do corpus de treinamento;
- Usar a informação do tipo erro anotado (err) para aprender regras, para depois usá-las na pós-edição com filtro;
- Eliminar do treinamento as subcategorias de erro palavra ausente e n-grama ausente, pois tais erros não foram tratados na pós-edição e podem ter trazido ruído ao processo de aprendizado de regras já que, de acordo com a Tabela 4.7, respectivamente, 1,57% e 8,33% das ocorrências de erros de tais subcategorias são ambíguas;
- Testar o aprendizado de regras da μ-TBL com diferentes parâmetros para precisão e pontuação;
- Alterar a forma de aplicar as regras, além da ordem nume, gener, verboc e wd. Algumas sugestões já foram dadas na seção 5.3 como possíveis experimentos futuros;
- Experimentar o aprendizado com outros *templates*, gerados com o auxílio de linguistas;
- Inserir regras criadas manualmente, como as de (AVANÇO; NUNES, 2013), nos arquivos de regras existentes;
- Testar a pós-edição com filtro usando também os modelos treinados com os algoritmos SVM e Naive Bayes, os quais, embora não tenham apresentado os melhores resultados na identificação automática de erros isoladamente, talvez tenham bom desempenho como passo prévio da correção.

Capítulo 6

CONCLUSÕES

Apesar de diferentes pesquisas realizadas até hoje em tradução automática (TA), os textos traduzidos automaticamente geralmente precisam ser pós-editados por tradutores humanos para que finalmente atinjam boa qualidade na língua alvo. Visando melhorar a saída de sistemas de TA e, ao mesmo tempo, diminuir a carga de trabalho do processo de pós-edição manual; nesta pesquisa, propôs-se investigar, implementar e avaliar métodos automáticos de pós-edição de textos traduzidos automaticamente (*Automated Post-Editing* ou APE) com o intuito de melhorar a qualidade desses textos. O resultado desta pesquisa é um APE implementado para corrigir erros gerados pela TA de textos em inglês para o português do Brasil, o EdiTA.

Para tanto, o processo de pós-edição foi dividido em duas etapas distintas porém relacionadas: (i) identificação automática de erros de TA e (ii) correção automática. Deste modo, foi possível investigar a pós-edição com identificação prévia do erro (usando filtro de incorretos ou por categorias, chamada de pós-edição com filtro) e sem identificação prévia do erro (pós-edição direta).

Os experimentos realizados para a identificação automática de erros de TA mostram que é possível atingir uma boa precisão (cerca de 77%) na classificação de um segmento traduzido por TA como correto com base em uma janela de 5 *tokens* (TW-5) e um conjunto pequeno de *features*[1] usando o algoritmo de aprendizado de máquina (AM) árvore de decisão. No entanto, o modelo quando aplicado ao corpus de teste apresenta uma queda na precisão (50%). Ainda usando o mesmo algoritimo de AM e tamanho de janela, a classificação em categorias de erros conseguiu uma boa precisão para erros sintáticos (cerca de 65%) e erros lexicais (cerca de 67%). Entretanto, não atingiu bons resultados para as demais categorias de erro. As melhores taxas de classificação foram obtidas pelas categorias de erro mais frequentes, ou seja, erros lexicais (44,48% das instâncias de treinamento) e sintáticos (38,61% das instâncias de treinamento).

[1] Como explicado no capítulo 4, o número de *features* varia dependendo do tamanho da TW. São 32 *features* para TW-5, 40 para TW-7 e 56 para TW-11.

Outro fator que poderia explicar os resultados ruins para a identificação automática de erros de algumas categorias e de todas as subcategorias é a sobreposição de erros anotados para uma mesma TW (veja Tabela 4.7). Essa sobreposição prejudica o aprendizado, já que nos modelos aprendidos cada trecho pode ter apenas uma classe (correto, incorreto, categoria ou subcategoria de erro) atribuída na identificação automática.

Acredita-se que a abordagem proposta pode levar a bons resultados na identificação automática de erros, desde que existam instâncias suficientes para o treinamento dos algoritmos de AM. O aumento na quantidade de instâncias de treinamento e consequentemente na diversidade de erros nas instâncias exigiria trabalho de anotadores humanos e demandaria um prazo considerável, por essa razão não foi possível aumentar o corpus anotado dentro do escopo de tempo deste projeto.

Além do aumento do corpus de treinamento, outras *features* podem ser investigadas em experimentos futuros, tais como:

- Uso de características obtidas a partir de árvores semânticas rasas (*shallow semantic trees*) descritas em (AZIZ; RIOS; SPECIA, 2011);
- Uso de relações semânticas como hiponímia e meronímia presentes nas instâncias (TABA; CASELI, 2012);
- Uso de características observáveis em árvores sintáticas da tradução em comparação com árvores sintáticas da sentença fonte (FELICE; SPECIA, 2012).

Com base nos resultados dos experimentos realizados para a primeira etapa do processo de pós-edição, a identificação de erros, acredita-se que as taxas de classificação obtidas permitam o envio de janelas classificadas com erros ao módulo de correção com um certo grau de confiança – tanto instâncias classificadas como incorretas como usando uma classificação mais refinada (em erros sintáticos e lexicais).

Considerando-se os trabalhos relacionados, uma pequena parte das *features* utilizadas no treinamento do identificador de erros apresentado neste documento, foi proposta por Felice e Specia (2012). A classificação usada, no entanto, foi diferente. Enquanto aqui realizou-se a classificação em corretos e incorretos, categorias ou subcategorias de erros, o classificador de Felice e Specia (2012) estima a qualidade da TA atribuindo uma nota de 1 a 5 dependendo do esforço de pós-edição. Na avaliação de sua pesquisa, Felice e Specia (2012) não obtiveram resultados melhores que o *baseline* (que utilizava outras *features*) e apontaram que talvez a aplicação de novas características linguísticas no modelo pudesse melhorar a classificação. Isso demonstra que a escolha do conjunto de *features* também influencia nos resultados alcançados e não somente o tamanho do conjunto de treinamento.

Além da identificação de erros, o EdiTA enbloba a etapa de correção automática da saída da TA em dois cenários: sem a etapa prévia de identificação automática (pós-edição direta) e com filtro de incorretos ou categoria de erros (pós-edição com filtro). Considerando-se apenas as medidas automáticas de avaliação geral da TA BLEU e NIST, não é possível concluir que a utilização do EdiTA melhore a qualidade da tradução gerada pelo *baseline*, o TAEIP. Contudo, com base em uma análise mais detalhada de precisão e cobertura das regras, bem como a análise manual das alterações por elas realizadas, conclui-se que o EdiTA tem um impacto positivo na saída do TAEIP.

Os experimentos demonstraram que na pós-edição direta o número de sentenças pós-editadas (51 no corpus de teste, de acordo com a análise manual) foi maior do que na pós-edição com filtro (16 ou menos no corpus de teste, de acordo com a análise manual). No entanto, em apenas 39% dos casos a alteração realizada pelo pós-editor direto trouxe melhora na saída da TA. Já no pós-editor com filtro, usando filtro de incorretos e categorias de erros, as precisões subiram para 69% e 100%, respectivamente. Esse fato confirma uma das hipóteses de trabalho desta pesquisa (H2): a identificação de erros como passo prévio à correção deve ser aplicada para evitar a geração de ruídos, ou seja, a alteração desnecessária de trechos corretos.

Vale notar, também, que a pequena quantidade de sentenças alteradas pela aplicação do EdiTA nos corpora de treinamento e de teste explica porque os valores de BLEU e NIST permaneceram praticamente inalterados. Além disso, a pequena quantidade de sentenças pós-editadas no corpus de teste (51, de acordo com a análise manual) em relação à quantidade de sentenças pós-editadas no corpus de treinamento (105, de acordo com a análise manual) é um indício de que o aprendiz de regras usando TBL é altamente dependente do corpus de treinamento (um indício de *overfitting*). Outra constatação obtida com a realização da verificação manual é a de que as medidas automáticas BLEU e NIST penalizaram o sistema quando comparadas com a avaliação feita por um humano, tanto na pós-edição com filtro como na sem filtro.

Por fim, é importante mencionar que o EdiTA está sujeito a ruídos (distorções nos resultados) em todas as etapas do processo, pois envolve o uso de várias ferramentas. Tais ruídos podem ocorrer, por exemplo, no alinhamento automático da sentença original (Src) com as versões traduzida automaticamente (Sys) e de referência (Ref). Como esse alinhamento desempenha papel fundamental na geração tanto de *features* para os algoritmos de aprendizado da primeira etapa (identificação de erros) como de *templates* para o TBL na segunda etapa (correção de erros), um alinhamento incorreto pode ser bastante prejudicial para o sistema resultante.

Embora os resultados obtidos nesta pesquisa apontem para uma direção promissora, experimentos adicionais precisam ser executados no futuro com o intuito de aumentar a qualidade da pós-edição. Algumas linhas que poderão ser investigadas são:

- Aumentar o tamanho do corpus de treinamento;
- Otimizar o treinamento utilizando um corpus de validação diferente do corpus de treinamento;
- Usar a informação sobre a categoria do erro anotado para aprender regras, para depois usá-las na pós-edição com filtro;
- Eliminar do treinamento as subcategorias de erro não tratadas na pós-edição, pois podem ter trazido ruído ao processo de aprendizado de regras;
- Testar o aprendizado de regras usando TBL com diferentes parâmetros;
- Alterar a ordem de aplicação das regras de correção;
- Experimentar o aprendizado de regras com outros *templates*, gerados com o auxílio de linguistas;
- Inserir regras criadas manualmente nos arquivos de regras existentes;
- Testar a pós-edição com filtro usando também os modelos treinados com os algoritmos SVM e Naive Bayes, os quais, embora não tenham apresentado os melhores resultados na identificação automática de erros isoladamente, talvez tenham bom desempenho como passo prévio da correção.

Dos trabalhos relacionados apresentados no Capítulo 2, três são facilmente comparáveis a este porque usaram o mesmo corpus, *baseline* ou técnica de AM. Em (UENISHI, 2013), realizou-se o treinamento de um APE estatístico para o TAEIP (mesmo *baseline* usado nesta pesquisa). O treinamento do APE estatístico (tradutor monolíngue de traduções ruins para boas traduções) foi executado com textos em português traduzidos automaticamente pelo TAEIP e os textos originais (gerados por humanos) em português presentes no corpus FAPESP-v2 (AZIZ; SPECIA, 2011). O APE treinado foi aplicado para pós-processar as saídas do TAEIP, mas os resultados, em termos de BLEU e NIST demonstraram uma queda na qualidade das sentenças após a pós-edição. Essa constatação também foi confirmada pela análise manual. Apenas uma parte do corpus FAPESP-v2 (teste-a) foi usada no treinamento do EdiTA, enquanto uma parte bem maior (todo o corpus menos teste-a e teste-b) foi usada no treinamento do APE estatístico de Uenishi. Desse modo, tem-se que a melhora conseguida pelo EdiTA, ainda que pequena, com um corpus de treinamento bem menor, para o mesmo *baseline* mostra que a abordagem utilizada neste projeto é promissora.

Em (AVANÇO; NUNES, 2013), regras manuais para corrigir erros de concordância nominal e verbal na saída do TAEIP foram criadas tomando como base o corpus teste-a (AZIZ; SPECIA,

2011) anotado manualmente com erros de TA (corpus de treinamento usado neste trabalho). Quando as regras manuais foram aplicadas houve um ganho de 2,85% em BLEU e 0,99% em NIST na avaliação de 35 sentenças para concordância nominal; e de 0,46% em BLEU e 0,29% em NIST na avaliação de 40 sentenças para concordância verbal. Ainda usando o TAEIP e o mesmo corpus, foram feitos testes também aplicando o verificador gramatical do MS-Word 2010 para a correção de erros de TA. Nesses experimentos houve um aumento de 1,14% em BLEU e 0,45% em NIST na avaliação de 35 sentenças para concordância nominal; e de 1,12% em BLEU e 0,37% em NIST na avaliação de 40 sentenças para concordância verbal. Apesar dos experimentos em (AVANÇO; NUNES, 2013) terem obtidos resultados melhores que os conseguidos pelo EdiTA, no projeto aqui apresentado o foco é diferente, pois um de seus objetivos é aprender automaticamente a identificar erros de TA e também aprender automaticamente regras de correção desses erros. Além disso, o escopo de sentenças avaliadas por BLEU e NIST nos experimentos do EdiTA foi bem maior em comparação com as sentenças avaliadas em (AVANÇO; NUNES, 2013).

Finalmente, Elming (2006) aplica a mesma técnica de AM (TBL) e a mesma ferramenta de aprendizado (μ-TBL) empregadas neste projeto para aprender regras de pós-edição da saída de um tradutor automático baseado em regras (RBMT). Em tal trabalho, utilizou-se um corpus de treinamento para o TBL com aproximadamente 12000 sentenças juntamente com um conjunto de validação de 2000 sentenças. Foram criados 70 *templates* baseados em etiquetas morfossintáticas e formas superficiais (palavras ou *tokens* da maneira como ocorrem no texto) para tratar da substituição de palavras – usando a influência contextual dos 6 *tokens* mais próximos, 3 de cada lado. Ao aplicar as regras aprendidas, Elming relata um aumento em BLEU, em relação ao desempenho do tradutor automátio RBMT sendo avaliado, de 59,5 para 63,5 usando o conjunto de teste.

Apesar das semelhanças em relação à estratégia escolhida para a implementação do EdiTA, uma comparação direta dos resultados desta pesquisa com os de (ELMING, 2006) não é possível pois: (i) o corpus de treinamento usado em (ELMING, 2006) é formado por sentenças da saída da TA pós-editadas manualmente e não de referência como nesta pesquisa garantindo, assim, um paralelismo maior entre tradução e versão correta , (ii) o tradutor *baseline* usado foi um RBMT e não um PB-SMT (estratégia considerada o estado da arte segundo BLEU e NIST) usado neste projeto, (iii) o corpus de treinamento usado por Elming é bem maior (12000 sentenças aproximadamente) do que o utilizado nesta pesquisa (com 1314 sentenças), (iv) Elming dispunha de 2000 sentenças para a validação na fase de treinamento enquanto a validação neste projeto foi realizada usando o próprio corpus de treinamento (devido ao tamanho limitado do corpus optou-se por não dividi-lo).

Assim, mesmo com a pós-edição do EdiTA não apresentando mudanças significativas nos

valores calculados pelas medidas BLEU e NIST, em comparação com os obtidos pelo *baseline*, esta pesquisa confirmou as hipóteses que se propôs a investigar pois, conforme verificado na análise detalhada de precisão e cobertura das regras e na análise manual: (H1) a pós-edição automática melhora a qualidade da TA e (H2) a identificação de erros realizada como passo prévio à correção evita que alterações desnecessárias sejam realizadas, ou seja, em H2 há menos alterações, mas tende-se a cometer menos erros, pois as pós-edições ocorrem apenas em trechos que realmente precisam ser alterados.

REFERÊNCIAS

ARMENTANO-OLLER, C. et al. Open-source Portuguese-Spanish machine translation. In: *Proceedings of the 7th International Workshop on Computational Processing of Written and Spoken Portuguese*. Itatiaia, RJ: [s.n.], 2006. p. 50–59.

AVANÇO, L. V.; NUNES, M. das G. V. *Pós-edição de traducções automáticas: avaliando a utilização de recursos disponíveis*. [S.l.], Outubro 2013.

AZIZ, W.; RIOS, M.; SPECIA, L. Shallow Semantic Trees for SMT. In: *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT 2011)*. Edinburgh, Scotland, UK: [s.n.], 2011. p. 316–322.

AZIZ, W.; SOUSA, S. C. M. de; SPECIA, L. PET: a Tool for Post-editing and Assessing Machine Translation. In: *Eighth international conference on Language Resources and Evaluation (LREC 2012)*. Istanbul, Turkey: [s.n.], 2012. p. 3982–3987.

AZIZ, W.; SPECIA, L. Fully Automatic Compilation of Portuguese-English and Portuguese-Spanish Parallel Corpora. In: *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology (STIL 2011)*. Cuiabá, MT, Brazil: [s.n.], 2011. p. 234–238.

BÉCHARA, H.; MA, Y.; GENABITH, J. van. Statistical post-editing for a statistical MT system. In: *Proceedings of the Thirteenth Machine Translation Summit (MT Summit XIII)*. Xiamen, China: [s.n.], 2011. p. 308–315.

BRILL, E. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. In: *Computational Linguistics*. [S.l.: s.n.], 1995.

BROWN, P. F. et al. A statistical approach to machine translation. *Computational Linguistics*, v. 16, n. 2, p. 79–85, 1990.

BROWN, R. D. Example-based machine translation in the Pangloss system. In: *Proceedings of the 16th International Conference on Computational Linguistics (COLING 1996)*. Copenhagen, Denmark: [s.n.], 1996. p. 169–174.

CALLISON-BURCH, C. et al. (Meta-) Evaluation of Machine Translation. In: *Proceedings of the Second Workshop on Statistical Machine Translation (ACL)*. Prague, Czech Republic: [s.n.], 2007. p. 136–158. Disponível em: <http://www.mt-archive.info/ACL-SMT-2007-Callison-Burch.pdf>. Acesso em: 30 out. 2012.

CANCEDDA, N. et al. Learning Machine Translation. In: ____. London, England: MIT, 2009. cap. 1, p. 3–23.

CASELI, H. M. *Indução de léxicos bilíngües e regras para a tradução automática (In Portuguese)*. Tese (Doutorado) — USP, São Carlos, São Paulo, 2007.

DODDINGTON, G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: *ARPA Workshop on Human Language Technology*. [s.n.], 2002. Disponível em: <http://www.itl.nist.gov/iad/mig//tests/mt/doc/ngram-study.pdf>.

DOYON, J. et al. Automated Machine Translation Improvement Through Post-Editing Techniques: Analyst and Translator Experiments. In: *8th AMTA conference*. Hawaii: [s.n.], 2008. p. 346–353.

ELMING, J. Transformation-based correction of rule-based MT. In: *EAMT*. [S.l.: s.n.], 2006.

FARRÚS, M. et al. Linguistic-based evaluation criteria to identify statistical machine translation errors. In: *Proceedings of EAMT*. Saint Raphael, France: [s.n.], 2010. p. 52–57. Disponível em: <http://www.mt-archive.info/EAMT-2010-Farrus.pdf>. Acesso em: 30 out. 2012.

FELICE, M.; SPECIA, L. Linguistic features for quality estimation. In: *Proceedings of the 7th Workshop on Statistical Machine Translation*. Montreal, Canada: [s.n.], 2012. p. 96–103.

FISHEL, M. et al. TerrorCat: a Translation Error Categorization-based MT Quality Metric. In: *Proceedings of the 7th Workshop on Statistical Machine Translation*. Montreal, Canada: [s.n.], 2012. p. 64–70.

GEORGE, C.; JAPKOWICZ, N. Automatic Correction of French to English Relative Pronoun Translations using Natural Language Processing and Machine Learning Techniques. In: *Computational Linguistics In the North East*. Ottawa, Canada: [s.n.], 2005.

GOMES, F. T.; PARDO, T. A. S. Trapezio – Translation Post Editor: um ambiente de pós-edição de traduções automáticas. In: *Anais do Congresso da Academia Trinacional de Ciências (C3N)*. Foz do Iguaçu, Paraná: [s.n.], 2008. p. 1–10. Disponível em: <http://www.icmc.usp.br/ taspardo/C3N2008-TassarioPardo.pdf>. Acesso em: 30 out. 2012.

HALL, M. et al. The WEKA Data Mining Software: An Update. *SIGKDD Exploration*, v. 11, p. 10–18, 2009. Disponível em: <http://www.kdd.org/explorations/issues/11-1-2009-07/p2V11n1.pdf>. Acesso em: 30 out. 2012.

HASTIE, T.; TIBSHIRANI, R. Classification by pairwise coupling. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 1998.

HOLTE, R. C. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, v. 11, p. 63–91, 1993.

HUTCHINS, J. Towards a definition of example-based machine translation. In: *Proceedings of MT Summit X*. [S.l.: s.n.], 2005. p. 63–70.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: *Eleventh Conference on Uncertainty in Artificial Intelligence*. San Mateo: [s.n.], 1995. p. 338–345.

KAWAMORITA, C.; CASELI, H. M. Memórias de Tradução: auxiliando o humano a traduzir. Trabalho apresentado no Encontro de Linguística de Corpus (ELC 2012). 2012.

KOEHN, P. et al. Moses: Open Source Toolkit for Statistical Machine Translation. In: *Proceedings of the ACL 2007 Demo and Poster Sessions*. Prague: [s.n.], 2007. p. 177–180.

KRANIAS, L.; SAMIOTOU, A. Automatic Translation Memory Fuzzy Match Post-Editing: A Step beyond Traditional TM/MT Integration. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. Lisbon, Portugal: [s.n.], 2004. p. 331–334. Disponível em: <http://www.mt-archive.info/LREC-2004-Kranias.pdf>. Acesso em: 30 out. 2012.

KRINGS, H. P. *Repairing Texts - Empirical Investigations of Machine Translation Post-Editing Processes*. [S.l.]: The Kent State University Press, 2001.

LAGARDA, A. L. et al. Statistical Post-Editing of a Rule-Based Machine Translation System. In: *Proceedings of NAACL HLT 2009*. Boulder, Colorado: [s.n.], 2009. p. 217–220.

LAGER, T. The u-tbl system: Logic programming tools for transformation-based learning. In: *Proceedings of the third international workshop on computational natural language learning*. Bergen: [s.n.], 1999.

LAGER, T. *The u-TBL system user's manual*. Version 0.9. [S.l.], 2000. Disponível em: <http://www.ling.gu.se/ lager/mutbl.html>.

LEVENSHTEIN, V. I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, p. 707–710, February 1966.

LLITJÓS, A. F. *Automatic Improvement of Machine Translation Systems*. Tese (Doutorado) — Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, July 2007. Disponível em: <http://www.mendeley.com/research/automatic-improvement-of-machine-translation-systems/>. Acesso em: 30 out. 2012.

LOHR, S. The age of big data. *The New York Times*, 2012. Disponível em: <http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html>. Acesso em: 21 jan. 2014.

LOPEZ, A. Statistical Machine Translation. *ACM Computing Surveys*, v. 40, n. 3, p. 8:2–8:49, August 2008.

MANGU, L.; BRILL, E. Automatic rule acquisition for spelling correction. In: *Proceedings of the Fourteenth International Conference on Machine Learning, ICML*. [S.l.: s.n.], 1997.

MARTINS, D. B. de J. et al. Annotating translation errors in brazilian portuguese automatically translated sentences: first step to automatic post-edition. In: *Corpus Linguistics*. Lancaster, UK: [s.n.], 2013. p. 189–192.

MITCHELL, T. M. *Machine Learning*. McGraw-Hill, 1997. Disponível em: <http://www.cs.cmu.edu/ tom/mlbook.html>. Acesso em: 30 out. 2012.

NIEßEN, S. et al. An evaluation tool for machine translation: Fast evaluation for machine translation research. In: *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC)*. Athens, Greece: [s.n.], 2000. p. 39–45.

O'BRIEN, S. Teaching Post-editing: A Proposal for Course Content. In: *Sixth EAMT Workshop "Teaching machine translation"*. Manchester, England: [s.n.], 2002. p. 99–106.

OCH, F. J. Minimum error rate training in statistical machine translation. In: *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL 2003)*. Sapporo, Japan: [s.n.], 2003.

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

OCH, F. J.; NEY, H. The alignment template approach to statistical machine translation. *Computational Linguistics*, p. 417–449, 2004. 30(4).

PAPINENI, K. et al. BLEU: a method for automatic evaluation of machine translation: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual meeting of the Association for Computational Linguistics (ACL 2002)*. Philadelphia: [s.n.], 2002. p. 311–318. Disponível em: <http://www.mt-archive.info/ACL-2002-Papineni.pdf>. Acesso em: 30 out. 2012.

PLATT, J. Advances in kernel methods - support vector learning. In: ____. [S.l.: s.n.], 1998. cap. Fast Training of Support Vector Machines using Sequential Minimal Optimization.

POPOVIC, M. Hjerson: An Open Source Tool for Automatic Error Classification of Machine Translation Output. *The Prague Bulletin of Mathematical Linguistics*, n. 96, p. 59–67, 2011.

POPOVIC, M.; BURCHARDT, A. From Human to Automatic Error Classification for Machine Translation Output. In: *Proceedings of the 15th Conference of the European Association for Machine Translation*. Leuven, Belgium: [s.n.], 2011. p. 265–272.

POPOVIC, M.; NEY, H. Word Error Rates: Decomposition over POS classes and Applications for Error Analysis. In: *Proceedings of the 2nd ACL 07 Workshop on Statistical Machine Translation (WMT 07)*. Prague, Czech Republic: [s.n.], 2007. p. 48–55.

POPOVIC, M.; NEY, H. Towards Automatic Error Analysis of Machine Translation Output. *Association for Computational Linguistics*, v. 37, n. 4, p. 658–688, March 2011.

POTET, M. et al. Preliminary Experiments on Using Users' Post-Editions to Enhance a SMT System. In: FORCADA, M. L.; DEPRAETERE, H.; VANDEGHINSTE, V. (Ed.). *Proceedings of the 15th conference of the European Association for Machine Translation (EAMT 2011)*. Leuven, Belgium: [s.n.], 2011. p. 161–168.

QUINLAN, J. R. Induction of Decision Trees. In: *Readings in Machine Learning*. [S.l.]: Morgan Kaufmann Publishers, 1990. Originally published in *Machine Learning* 1:81–106, 1986.

QUINLAN, J. R. *C4.5*: Programs for Machine Learning. San Mateo,CA: Morgan Kaufmann Publishers, 1993.

SANTOS, C. N. *Entropy Guided Transformation Learning*. Tese (Doutorado) — PUC - Rio de Janeiro, March 2009.

SENEFF, S.; WANG, C.; LEE, J. Combining linguistic and statistical methods for bi-directional english chinese translation in the flight domain. In: *Proceedings of the AMTA*. [S.l.: s.n.], 2006.

SILVA, B. C. D. *Processamento Automático de Línguas Naturais*. Araraquara, March 2010. Apostila.

SIMARD, M.; GOUTTE, C.; ISABELLE, P. Statistical Phrase-based Post-editing. In: *Proceedings of NAACL HLT 2007*. Rochester, NY: [s.n.], 2007. p. 508–515.

SNOVER, M. et al. A study of translation edit rate with targeted human annotation. In: *Proceedings of AMTA*. Cambridge, Massachusetts, USA: [s.n.], 2006. p. 223–231. Disponível em: <http://www.mt-archive.info/AMTA-2006-Snover.pdf>. Acesso em: 30 out. 2012.

SPECIA, L. Exploiting objective annotations for measuring translation post-editing effort. In: *Proceedings of the 15th Conference of the European Association for Machine Translation*. Leuven: [s.n.], 2011. p. 73–80.

STOLCKE, A. SRILM an Extensible Language Modeling Toolkit. In: *Proceedings of the International Conference on Spoken Language Processing*. [S.l.: s.n.], 2002.

STYMNE, S. BLAST: A Tool for Error Analysis of Machine Translation Output: A Tool for Error Analysis of Machine Translation Output. In: *Proceedings of the ACLHLT 2011 System Demonstrations*. Portland, Oregon: [s.n.], 2011. p. 56–61.

STYMNE, S. Pre- and Postprocessing for Statistical Machine Translation into Germanic Languages. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Student Session*. Portland, Oregon: [s.n.], 2011. p. 12–17.

STYMNE, S.; AHRENBERG, L. Using a Grammar Checker for Evaluation and Postprocessing of Statistical Machine Translation. In: *Proceedings of the seventh international conference on Language Resources and Evaluation (LREC 2010)*. Valletta, Malta: [s.n.], 2010. p. 2175–2181.

TABA, L. S.; CASELI, H. M. Automatic Hyponymy Identification from Brazilian Portuguese Texts. In: *Lecture Notes in Artificial Intelligence*. Coimbra, Portugal: [s.n.], 2012. (International Conference on Computational Processing of Portuguese – PROPOR, v. 7243), p. 186–192.

TILLMANN, C. et al. Accelerated DP based search for statistical translation. In: *European Conf. on Speech Communication and Technology*. Rhodes, Greece: [s.n.], 1997. p. 2667–2670.

UENISHI, A. T. Trabalho de Conclusão de Curso, *Pós-edição automática estatística da saída da tradução automática*. São Carlos: [s.n.], 2013. 35 p.

VILAR, D. et al. Error analysis of statistical machine translation output. In: *Proceedings of the fifth International Conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy: [s.n.], 2006. p. 22–28.

ZEMAN, D. et al. Addicter: What Is Wrong with My Translations? *The Prague Bulletin of Mathematical Linguistics*, v. 96, n. 96, p. 79–88, October 2011.