5
Criando o Corpus Anotado

Os algoritmos utilizados para resolver a tarefa de anotação morfossintática dos *tweets* em português são de aprendizado supervisionado. Sendo assim, eles aprendem a partir de exemplos, que são passados a eles na forma de um corpus anotado. Os exemplos são *tweets*, isto é, mensagens postadas no Twitter por usuários. O corpus deve ser uma amostra representativa do universo de *tweets* e prover características suficientes de seus dados para que o classificador possa extrair conhecimento dele. Se o corpus tem erros de anotação, eles introduzem ruído no aprendizado, dificultando esta tarefa.

A geração de um corpus anotado é dividida em duas etapas principais: criação do corpus através da obtenção e *tokenização* de *tweets* representativos; e anotação dos *tweets*. Cada uma dessas etapas pode ser dividida em subetapas, como mostraremos a seguir.

5.1
Criando o Corpus

A etapa de criação do corpus deve gerar como saída uma lista de *tweets*, representativos do universo de *tweets* escritos em português. Cada um destes *tweets* deve estar *tokenizado*, para facilitar o processamento nas etapas seguintes. Para gerar a saída requerida, dividimos essa tarefa em 5 subtarefas, a saber: capturar os *tweets* a partir do Twitter; *tokenizar* os *tweets*; selecionar os *tweets* escritos em português; balancear os *tweets* selecionados; e normalizar o corpus, desfazendo combinações, contrações e composições de palavras.

Figura 5.1: Pipeline de geração do corpus.

A seguir, discutiremos cada subetapa que compõe a criação de um corpus, junto com seus desafios e como cada um foi transposto.

5.1.1 Obtendo tweets

Para obter os *tweets*, criamos um *crawler*. Um *crawler* é um software desenvolvido para varrer a internet de maneira sistemática, a fim de colher informações relevantes a sua função. Em nosso caso, ele percorre o Twitter em busca de *tweets*. O pseudo-código 5.1 ilustra um exemplo de um *crawler* simples.

Código 5.1: Pseudo-código de um *crawler* criado para capturar *tweets*.

```
obtem_tweets(usuarios_novos:[], usuarios_vistos = [], tweets = []):
    enquanto (usuarios_novos.tamanho() > 0):
        usuario = usuarios_novos.remove_primeiro()
        se (usuarios_vistos nao contem usuario):
            usuarios_vistos += [usuario]
            usuarios_novos += usuario.pega_amigos()
            tweets += usuario.pega_tweets(5)
```

O Twitter disponibiliza uma API REST para que aplicações possam acessá-lo com todas as funcionalidades que um usuário da web tem. Existem diversas implementações dessa API e em diversas linguagens [16]. Adotamos a JTwitter pois ela é bem difundida e fácil de usar, tendo em vista ser escrita em JAVA.

O Twitter permite que sejam feitas apenas 150 requisições por hora por meio desta API [17]. Para aumentar esse número e acelerar o *crawler*, nos cadastramos no Twitter [18], requisitando a nossa entrada em sua *whitelist*, que é uma lista de usuários autorizados a fazerem até 20.000 requisições por hora.

5.1.2 Tokenização

Tokenização é a tarefa de segmentar um texto em elementos linguísticos chamados de *tokens*. O método mais simples de *tokenizar* um texto é utilizar seus espaços em branco, quebra de linha e tabulação como separadores de *tokens*. Essa não é uma boa solução, pois não separa as palavras das pontuações. Por exemplo, a *tokenização* da frase "*Olá mundo!*" gera os *tokens* "*Olá*" e "*mundo!*".

A solução que empregamos é a de criar um *tokenizador* que segmenta a frase utilizando expressões regulares que identificam elementos significativos. A seguir, descrevemos os elementos a serem identificados.

Usuário - É uma sequência de caracteres iniciada por um @, separada do restante do texto por espaço, quebra de linha, tabulação ou pontuação.

"E ai @renata, q q vc tem feito???"

Tópico - É uma sequência de caracteres iniciada por um #, separada do restante do texto por espaço, quebra de linha, tabulação ou pontuação.

"Arrumar bolsa que afinal #carnaval acabou e hj tem academia"

"ele passo mais vai deixar saudades... #carnaval"

URL - Endereço de um site na web.

"estou votando no melhor do ano no site do domigão do fautão!! site www.globo.com/faustao!!!"

Abreviações - Substituição da forma plena de um vocábulo pela forma reduzida [19].

"vo toma banho p/ sai c/ a dra. @renata!!!"

E-mail - Endereço do correio eletrônico de um usuário.

"me manda um email q depois eu t respondon dummy@gmail.com!!!"

Moeda - Símbolo de moedas corrente.

"Vai t custar R$100,00"

Ênclise e mesóclise - Refere-se a colocação pronominal, que é a posição em que os pronomes pessoais oblíquos átonos ocupam na frase em relação ao verbo a que se referem. Os pronomes oblíquos átonos são: me, te, se, o, os, a, as, lhe, lhes, nos e vos. Na ênclise, o pronome vem depois do verbo, separado dele por um hífen.

"Diga-lhe que está tudo bem."

Na mesóclise, o pronome vem no meio do verbo.

"Far-lhe-ei uma proposta irrecusável."

Para classificar tanto o verbo quanto o pronome, o *tokenizador* os separa. Existe também a próclise, que indica que o pronome se encontra antes do verbo.

"Nada me faz querer sair dessa cama."

Como se pode observar no exemplo acima, o verbo e o pronome não se unem para formar uma nova palavra, então este caso já é tratado pela expressão que identifica **palavras**.

Emoticon - Forma de comunicação paralinguística, um *emoticon* é uma sequência de caracteres tipográficos que tem como objetivo expressar o estado emotivo de quem os emprega.

"Eu estou mega feliz :D"

Palavras - Vocábulo representado graficamente [19] que não se encaixa nas classes acima.

"@Maria, vc pagou 300 pratas ao ernesto pelos convites? :/"

Pontuação- Sinais gráficos que dividem as partes do discurso que não têm entre si ligação íntima [19].

"Uhuuu! Isso foi d+..."

Demais caracteres - Este tipo captura o que não se encaixa em nenhuma das categorias anteriores e não for espaço, quebra de linha ou tabulação.

"Na hora sai uns 10% mais barato"

5.1.3 Selecionando Exemplos Escritos Em Português-Twitter

Dada uma lista de *tweets tokenizados*, devemos selecionar os que são escritos em português-twitter. Para isto, basta classificar cada *tweet* como sendo escrito em português-twitter ou não, para depois eliminar os que não são.

Construímos um classificador de *tweets* utilizando o algoritmo Naïve Bayes. O corpus de treino é composto pelas línguas originais das quais os respectivos dialetos do Twitter se originam. O Naïve Bayes calcula a probabilidade de um *tweet* estar em uma dada língua como sendo o produtório das probabilidades de cada um de seus *tokens* estar naquela língua, ponderada pela probabilidade da língua no corpus, isto é,

$$Pr[lang|token_1, token_2, ..., token_n] = Pr[lang] * \prod_{i=1}^{n} Pr[token_i|lang]$$

onde $token_i$ é o i-ézimo dos n *tokens* daquele *tweet* e *lang* é a língua para qual se quer determinar a probabilidade. A probabilidade da linguagem é o total de entradas daquela língua no corpus de treino, dividido pelo tamanho do corpus de treino, isto é,

$$Pr[lang] = \frac{C_{lang}}{C}$$

onde C é o total de entradas do corpus e C_{lang} é o total de entradas em uma determinada língua.

A probabilidade de um *token* estar em uma determinada língua se calcula dividindo o número de vezes que aquele *token* apareceu em alguma entrada

daquela língua, pelo número de *tokens* presentes em todas as entradas daquela língua, isto é,

$$Pr[token|lang] = \frac{lang_{token}}{lang_{tokens}}$$

onde $lang_{tokens}$ é o total de *tokens* presentes nas entradas daquela língua e $lang_{token}$ é o número de vezes que que um determinado *token* apareceu nas entradas daquela língua.

Para evitar probabilidades 0, adotamos a suavização de Laplace, obtendo

$$Pr[token|lang] = \frac{lang_{token} + 1}{lang_{tokens} + lang_{voc}}$$

onde $lang_{voc}$ é o tamanho do vocabulário daquela língua.

Uma vez calculadas as probabilidades, basta atribuirmos a cada *tweet* a língua do Twitter correspondente a língua original que teve a maior probabilidade.

Isso funciona por que o Naïve Bayes considera cada *token* individualmente, ignorando a estrutura do texto. Ele funciona como um saco de palavras, no qual a presença de uma palavra no saco é independente das outras. Como o léxico de toda língua original está sempre contido no léxico de sua versão Twitter, existem boas chances dos *tweets* conterem palavras de sua língua de origem. Consequentemente, reconhecendo a língua de origem, podemos inferir em qual língua do Twitter ele foi escrito.

Uma vantagem desse processo é que tendo a probabilidade da classificação, nos preocupamos em validar manualmente apenas os que estejam abaixo de um limiar, ou seja, os que temos pouca confiança.

Também utilizamos essa saída corrigida para realimentar o classificador bayesiano ingênuo, aumentando a cada interação, a sua qualidade.

5.1.4 Balanceamento dos Exemplos

Os exemplos são balanceados, mantendo as proporções de usuários a partir de um estudo estatístico dos usuários brasileiros do Twitter. Um estudo desses, também chamado de demografia do Twitter, pode ser encontrado na referência bibliográfica [20].

Escolhemos balancear os exemplos considerando a faixa etária dos seus autores. Para isso, o *crawler* foi alterado para balancear os exemplos, selecionando usuários de maneira a manter as proporções indicadas na Tabela 5.1.

Idade	Frequência (%)
00 a 17	38
18 a 24	26
25 a 34	17
35 a 44	12
45 a 54	6
55 a 64	-
65 ou mais	-

Tabela 5.1: Percentual de brasileiros no TWITTER de acordo com a faixa etária.

O TWITTER não requisita este dado a seus usuários em momento algum, porém é muitas vezes encontrado no campo *biografia* do *profile* do usuário. Utilizamos uma expressão regular para obter essa informação da biografia do usuário.

5.1.5 Normalizando o Corpus

Com o intuito de simplificar o corpus e a criação do modelo, na etapa de normalização do corpus, desfazemos as ligações de preposições com outras palavras, formando vocábulos únicos. Para isto, utilizamos um dicionário. O conjunto de contrações e combinações para o português é finito. Considerando que o léxico do português está contido no léxico do português-twitter, o conjunto de contrações e combinações do português também está contido no conjunto de contrações e combinações do português-twitter. Consideramos neste conjunto, os vocábulos gerados por variações na escrita das contrações e combinações do português. Por exemplo, no português-twitter temos as contrações, de mesma semântica, *daqui* e *daki*. A única composição observada nos exemplos obtidos é a palavra *né*, que é então decomposta como *não* seguido de *é*. No Apéndice A.1, listamos as 120 contrações e combinações observadas no português-twitter.

5.2 Anotando os exemplos

Anotar um corpus é classificar seus exemplos, para uma determinada tarefa, com um determinado conjunto de etiquetas. Em nosso caso, anotamos o corpus, gerado na etapa anterior, para a tarefa de anotação morfossintática. O conjunto de etiquetas utilizadas é uma versão simplificada daquele utilizado no corpus MAC-MORPHO [9]. A diferença é a ausência das etiquetas complementares, bem como as de contrações, ênclises e mesóclises. Esse *tagset* foi

utilizado como entrada para outros classificadores mais complexos, tais como *phrase chunking* [7] e *named entity recognition* [8], demonstrando assim sua qualidade. A Tabela 5.2 apresenta as etiquetas utilizadas.

Classe Gramatical	*Etiqueta*
adjetivo	ADJ
advérbio	ADV
advérbio conectivo subordinativo	ADV-KS
advérbio relativo subordinativo	ADV-KS-REL
artigo	ART
conjunção coordenativa	KC
conjunção subordinativa	KS
interjeição	IN
substantivo	N
nome próprio	NPROP
numeral	NUM
particípio	PCP
palavra denotativa	PDEN
preposição	PREP
pronome adjetivo	PROADJ
pronome conectivo subordinativo	PRO-KS
pronome pessoal	PROPESS
pronome relativo conectivo subordinativo	PRO-KS-REL
pronome substantivo	PROSUB
verbo	V
verbo auxiliar	VAUX
símbolo de moeda corrente	CUR

Tabela 5.2: Etiquetas para anotação POS do português-twitter.

Elementos específicos do português-twitter, não encontrados no português, foram classificados de acordo com nossas necessidades, com a ajuda de um linguista[1], como descrito a seguir:

Usuário - por representar o nome de um usuário no Twitter, é classificado como um nome próprio, recebendo a etiqueta *NPROP*.

Tópico - pode ser utilizado para enumerar os tópicos presentes em um texto. Neste caso, é classificado como um substantivo, recebendo a etiqueta *N*.

[1]comunicação pessoal feita por Maria Cláudia de Freitas

Esta é a sua classificação padrão. Porém, muitas vezes ele é utilizado substituindo palavras no meio do texto. Neste caso, ele assume a classe da palavra substituída.

Retweet - é uma citação a um *tweet* postado anteriormente por alguém. Por chamar a atenção ao texto referido, é classificado como interjeição, recebendo a etiqueta *IN*.

URL e E-mail - endereços são classificados como substantivos, recebendo a etiqueta *N*.

Emoticons - expressam o estado de espírito do usuário no contexto daquele *tweet*, recebendo a etiqueta *IN*.

A maneira mais simples de anotar um corpus é através da anotação manual. Este processo demanda tempo e muito recurso humano especializado, o que é oneroso. Visando agilizar este processo e reduzir a quantidade de anotadores necessários, usualmente é utilizado um classificador automático simples. A seguir, a tarefa de revisar e corrigir essas anotações iniciais é atribuída a um grupo de anotadores especializados.

Desenvolvemos um processo de anotação que gera uma classificação inicial com excelente acurácia, reduzindo ainda mais o tempo necessário, a quantidade de anotadores e o seu respectivo conhecimento linguístico do português.

5.2.1 Classificador Inicial

Para o português, temos acesso a um Anotador Morfossintático de boa acurácia. Suponhamos que tivéssemos um processo de tradução automático e de boa qualidade, que convertesse *tweets* em português para o português usual. Neste caso, a tarefa de anotação morfossintática desses *tweets* teria uma solução trivial, a saber: aplicar o Tradutor para português seguido de um Anotador Morfossintático de português. Todavia, a sub-tarefa de tradução automática requer o conhecimento detalhado dos aspectos morfossintáticos e semânticos das duas linguagens, a origem e a alvo. Assim, é claramente uma tarefa mais difícil do que a simples análise morfossintática de uma única linguagem. Portanto, o desempenho dos tradutores é tipicamente inferior ao dos Anotadores Morfossintáticos. Desta forma, a baixa qualidade da tradução já limitaria a qualidade final da anotação morfossintática baseada numa tradução preliminar.

Por outro lado, a tarefa manual ou semi-automática de traduzir os *tweets* para o português, é mais simples do que anotar esses *tweets* com suas etiquetas morfossintáticas. Isso por que, na falta de uma gramática do português-twitter, o anotador primeiro traduz a frase para o português para depois gerar sua etiqueta morfossintática.

Assim, nossa estratégia para construir um classificador inicial consiste em cinco passos, a saber:

1) traduzir os *tweets* para português;

2) criar um mapeamento bidirecional entre os *tokens* dos *tweets* e suas respectivas traduções;

3) aplicar o anotador de POS nos *tweets* traduzidos e revisados;

4) mapear o POS dos *tweets* traduzidos e revisados para a correspondente versão original em português-twitter;

5) aplicar regras para anotar *tokens* especiais que não podem ser traduzidos.

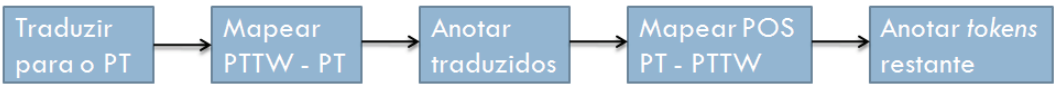

Figura 5.2: Pipeline de anotação do corpus.

Traduzindo do Português-Twitter para o Português

Para resolver à primeira sub-tarefa da anotação do corpus, desenvolvemos um tradutor semi-automático. Esse tradutor é composto por uma sequência de seis passos.

Levando em conta que o léxico do português está contido no léxico do português-twitter, o primeiro passo consiste em simplesmente replicar o *tweet*.

No segundo passo, é utilizado um dicionário para traduções simples, como as das abreviações, por exemplo, “vc” vira “você” e “q” vira “que”.

No terceiro passo, são utilizadas regras básicas, escritas manualmente, para fazer as seguintes correções:

- Se o *token* é um e-mail, substituir pela palavra “e-mail”.
- Se o *token* é uma *url*, substituir pela palavra “site”.
- Se o *token* é um *emoticon* - conjunto de caracteres predominantemente não alfanuméricos -, substituir por vazio.

- Se o *token* é um *retweet* - "RT" -, substituir por vazio.
- Se o *token* é uma risada - "Rs", ou conjunto de 4 ou mais caracteres onde a letra 'H' predomina -, substituir por vazio.
- Se o *token* é o primeiro *token* válido e é uma pontuação, substitui por vazio.
- Se o *token* a ser traduzido é o primeiro *token* válido, ou o último a ser avaliado e representa um usuário - começa com @ -, substituir por vazio.
- Se o *token* é um usuário - começa com @ - e o *token* seguinte e/ou anterior também, substituir por vazio.
- Se o *token* a ser traduzido é o primeiro *token* válido, ou o último a ser avaliado e representa um tópico - começa com # -, substituir por vazio.
- Se o *token* é um tópico - começa com # - e o *token* seguinte e/ou anterior também, substituir por vazio.
- Se o *token* não é uma risada e termina com a letra 'h' precedida de uma vogal, substituir ambas as letras por aquela vogal com um acento agudo.

As regras que transformam usuários e tópicos em vazio podem parecer estranhas a princípio, mas examinando o corpus, é possível observar que na maioria dos casos descritos pelas regras acima, esses não fazem parte da mensagem sendo transmitida pelo *tweet*. No caso dos usuários, eles existem apenas para chamar a atenção daqueles usuários para o *tweet*. No caso do tópico, marcar os assuntos presentes no *tweet*.

O quarto passo é a correção automática da caixa das palavras. No português-twitter a caixa das palavras pode ser utilizada para expressar emoção. Em decorrência disso, a caixa deve ser corrigida. A estratégia utilizada é transformar todas as palavras para caixa baixa, com exceção da primeira letra da primeira palavra e da primeira letra de cada palavra precedida por uma elipse, um ponto final, um ponto de exclamação, ou um ponto de interrogação. Palavras com até 4 caracteres em caixa alta, rodeadas de palavras escritas em caixa baixa, são consideradas siglas e por isso sua caixa alta é mantida.

O quinto passo tem como objetivo reutilizar o conhecimento aprendido em cada tradução. Para isso, utilizamos um classificador unigrama, que começa vazio e tem seu modelo atualizado a cada saída da etapa tradução. Consequentemente, se ao traduzir um determinado *tweet* aprendemos que '10pejo' se traduz para 'despejo', esse conhecimento é utilizado para traduzir automaticamente essa palavra em *tweets* subsequentes.

O último passo envolve a revisão por humanos das traduções geradas automaticamente. Esta tarefa, apesar de enfadonha, é trivial para qualquer

humano fluente em português. Sendo assim, tem baixo custo e é executada com bastante qualidade e eficiência. A saída dela realimenta o classificador por frequência.

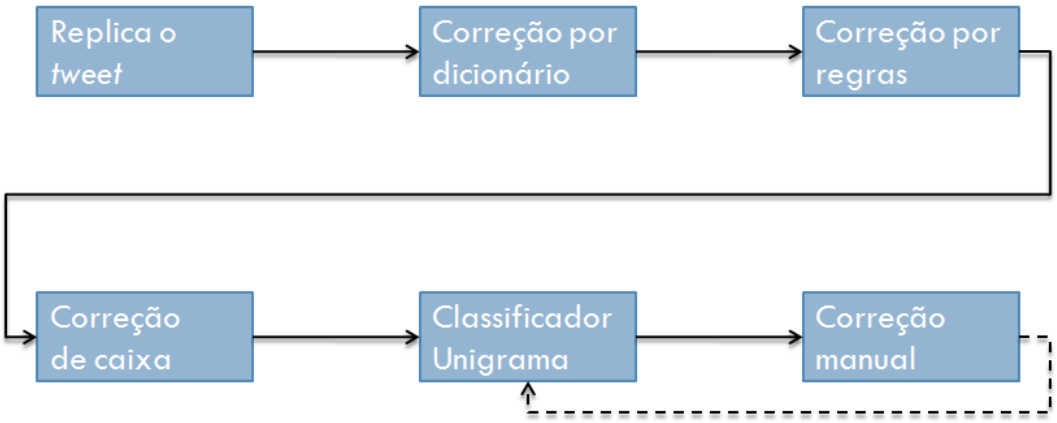

Figura 5.3: Pipeline do tradutor semi-automático.

A Figura 5.4 mostra um exemplo de como a tradução evolui neste processo. Neste exemplo, o classificador não conhece a palavra "participa", mas já conhece a palavra "entra" e sabe traduzi-la para "entrar".

tweet	RT @Pedro: p/ participa eh soh vc entra em dummy.com
Tradução por dicionário	RT @Pedro: para participa eh soh você entra em dummy.com
Correção por regras básicas	* * * para participa é só você entra em site
Correção de caixa	Para participa é só você entra em site
Classificador unigrama	Para participa é só você entrar em site
Correção manual	Para participar é só você entrar em o site

Figura 5.4: Exemplo de tradução do português-twitter para o português.

Mapeando entre o Português-Twitter e o Protuguês

O mapeamento entre os *tokens* dos *tweets* com suas respectivas traduções é um subproduto do processo de tradução. Assim, o segundo passo do processo de anotação é trivial. Basta manter referências entre os *tokens* originais e suas respectivas traduções em cada *tweet*, durante o processo de tradução.

PT-TW	@Pedro			q		q	vc	q	faze	hj	:P	?
PT	Pedro	,	o	que	é	que	você	quer	fazer	hoje		?

Figura 5.5: Exemplo de mapeamento entre uma frase escrita em português-twitter e sua tradução para o português.

Anotando Tweets Traduzidos para o Protuguês

Para o terceiro passo, utilizamos o Anotador Morfossintático para o português desenvolvido pelo laboratório Learn [2] e disponibilizado através do *web service* F-EXT-WS [21]. Este Anotador Morfossintático para português tem acurácia de 96,6%, sendo adotado então para anotar os *tweets* traduzidos e revisados.

5.2.2 Transferir o POS dos Tokens Traduzidos para o Original

Os passos anteriores, de tradução, mapeamento e anotação do corpus traduzido, visam facilitar a anotação do POS para o português-twitter. Abaixo, mostramos como é feita essa anotação a partir dos itens gerados: corpus traduzido e anotado com o POS e mapa entre os *tokens* do corpus original e do traduzido.

PT-TW	@fe/?	A/?	negação/?	eh/?	a/?	primeira/?	fase/?	
PT		A/ART	negação/N	é/V	a/ART	primeira/ADJ	fase/N	./.

Figura 5.6: Exemplo de mapeamento entre *tweet* escrito em português-twitter e sua versão traduzida e anotada.

No quarto passo, utilizando o mapa gerado na Seção 5.2.1, transferimos o POS anotado no corpus traduzido, para o corpus original, escrito em português-twitter.

PT-TW	@fe/?	A/ART	negação/N	eh/V	a/ART	primeira/ADJ	fase/N	
PT		A/ART	negação/N	é/V	a/ART	primeira/ADJ	fase/N	./.

Figura 5.7: Transferindo o POS da versão traduzida para o *tweet* escrito em português-twitter.

[2] http://www.learn.inf.puc-rio.br/

Anotar Tokens Restantes

O quinto passo no processo de anotação dos *tweets* consiste em anotar alguns *tokens* que não podem ser traduzidos no contexto em que aparecem. Consequentemente, eles não são anotados com o POS na etapa anterior. Felizmente, esse *tokens* são elementos do TWITTER, fáceis de serem identificados. Eles são anotados com sua categoria padrão, definidas na Seção 5.2.

PT-TW	@fe/NPROP	A/ART	negação/N	eh/V	a/ART	primeira/ADJ	fase/N	
PT		A/ART	negação/N	é/V	a/ART	primeira/ADJ	fase/N	./.

Figura 5.8: Anotando elementos que faltam com sua categoria padrão.

5.2.3 Revisar o Corpus Anotado

Após todas essas etapas, para garantir a qualidade do corpus, é necessária a revisão do mesmo por um especialista, um linguista.

PT-TW	@fe/NPROP	A/ART	negação/N	eh/V	a/ART	primeira/NUM	fase/N	
PT		A/ART	negação/N	é/V	a/ART	primeira/ADJ	fase/N	./.

Figura 5.9: Revisando e corrigindo o *tweet* escrito em português-twitter.

Entretanto, essa tarefa é menos dispendiosa que a tarefa inicial de anotar o do zero. Ela também é menos suscetível a erro, pois grande parte do corpus é anotada a partir de um anotador com acurácia de 96%.

5.3 Estatísticas do Corpus

A partir do processo descrito acima, foram criados dois corpus, um corpus de teste e um corpus de treino.

O corpus de treino é composto de 39.686 sentenças, contendo 616.760 *tokens*, dando uma média de 15,54 *tokens* por sentença e um vocabulário de 63.108 palavras.

O corpus de teste é composto de 5.490 sentenças, contendo 87.521 *tokens*, dando uma média de 15,94 *tokens* por sentença e um vocabulário de 15.389 palavras.

Na anotação do corpus de teste, executamos integralmente a etapa de revisão da tradução. Além disso, as correspondentes etiquetas de POS

estão superficialmente revisadas por um linguista[3]. Sendo assim, atribuímos a esta anotação um grau de confiança equivalente a acurácia do ANOTADOR MORFOSSINTÁTICO automático utilizado, isto é, 96,6%.

O corpus de treino foi traduzido usando o conhecimento aprendido na tradução do corpus de teste. Ele não pode ser completamente revisado devido ao seu tamanho e a falta de recursos. Isso poderá acarretar erros nos modelos gerados com esse corpus. Entretanto, acreditamos que o tamanho do corpus irá reduzir a taxa de erro.

[3]Tiago da Silva Ribeiro