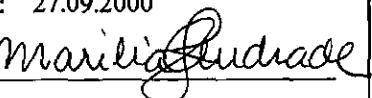


SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 27.09.2000

Assinatura: 

Implementação, Adaptação, Combinação e Avaliação de Etiquetadores para o Português do Brasil

Rachel Virgínia Xavier Aires

Orientadora: *Profa. Dra. Sandra Maria Aluísio*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Área: Ciências de Computação e Matemática Computacional.

USP – São Carlos
Setembro de 2000

Última flor do Lácio, inculta e bela,
És, a um tempo, esplendor e sepultura;
Ouro nativo, que, na ganga impura,
A bruta mina entre os cascalhos vela...

Amo-te assim, desconhecida e obscura,
Tuba de alto clangor, lira singela,
Que tens o trom e o silvo da procela,
E o arrolo da saudade e da ternura!

Amo o teu viço e o teu aroma
De virgens selvas e de oceanos largos!
Amo-te, ó rude e doloroso idioma,
Em que da voz materna ouvi: "meu filho!"
E em que Camões chorou, no exílio amargo,
O génio sern ventura e o amor sem brilho!

Língua Portuguesa - Olavo Bilac

Aos mestres da minha vida — vózinha, vovô Diana, vó Cog, vó Vitor, vó Xavier, mamãe, papai, nana, tia Cordélia, Dinda, Dada, tio Vitor, tios, tias, priminhos e priminhas —, que a mim dedicaram tantos olhares.

AGRADECIMENTOS

Tentar tornar-se mestre em uma determinada área é uma tarefa sem dúvida alguma cansativa. No entanto, ao longo destes dois anos, com muita ajuda, o que era um simples mestrado em computação, transcendeu esta conotação e se tornou delicioso e enriquecedor. Parecia uma conspiração do Universo para que tudo desse certo. Em 1998 lá estava eu fazendo mestrado na USP, na área dos meus sonhos, já com alguns amigos, é claro que em um vilarejo, mas nada é perfeito, nem mesmo uma conspiração do Universo. É por tudo ter dado tão certo, quer dizer, quase tudo, que sou grata por tudo que aconteceu e a todos que passaram pela minha vida nestes anos. Em especial agradeço:

- ☺ a Deus por ter traçado para mim um lindo caminho, colocando a natureza a minha frente quando necessário para me tranquilizar, permitindo que nunca perdesse a esperança;
- ☺ aos meus pais e à minha maninha pelo colo, pelo carinho que conforta e dá ânimo;
- ☺ aos amigos de Goiânia que não me esqueceram apesar da distância, em particular a amizade incondicional da Ludi, Ceci, Érika e Túlius;
- ☺ à família do Labic que me acolheu com todo carinho;
- ☺ ao prazer de ter uma orientadora amiga;
- ☺ aos amigos de Goiânia que vieram pra cá na mesma época e compartilharam das mesmas saudades — Katinha e Cláudio;

- ☺ aos amigos que me apresentaram a USP e a cidade - Nilda, Luiz Carlos, Marco, Ernesto (Tio) e Paulo;
- ☺ aos amigos bons ouvintes que fiz aqui como o Má, a Ângela e a Dê;
- ☺ à torcida fofoíssima do Alan, da Gio e do Le;
- ☺ aos amigos do Nilc;
- ☺ aos demais amigos, mas não menos importantes — Ana Paula, Mariane, Bléia, Chavas, Marquinhos, Daniele, Rosália, Calebe, Xico, Du, Adriano e Leo;
- ☺ ao apoio moral do Chú;
- ☺ às fadas e magos do português do Nilc que de alguma forma trabalharam neste projeto — Ana Cláudia, Denise, Gisele, Raquel e Ronaldo;
- ☺ ao Márcio, um aluno de inciação científica fabuloso, que trabalhou em finais de semana e nas férias com a maior boa vontade (e a Tati por não ter se importado com tanta hora-extra);
- ☺ ao King Ma por ter fornecido o código do etiquetador neural elástico;
- ☺ à Eric Brill, Helmut Schimid e a todos da lista corpora@hd.uib.no por terem tirado minhas dúvidas por email
- ☺ ao superapoio técnico dos amigos: Walter, Robson, Cláudio Hirosi, IA, Ju (dona da BDL) e Renatinho;
- ☺ à ajuda da Christie, Mônica e Ariane que foram diversas vezes importunadas com perguntas do tipo isso escreve assim, assim fica mais bonito, essa ou essa fonte;
- ☺ à ajuda da Daniela que tirou várias dúvidas mesmo me ouvindo falar tão mal de São Carlos;
- ☺ à ajuda da mica Andréia com redes neurais;
- ☺ à ajuda com estatística dos professores Jorge e Creusa;
- ☺ às conversas com a professora Carolina sobre classificadores e técnicas estatísticas de avaliação;
- ☺ ao apoio financeiro do CNPQ e da Intelligenesis;
- ☺ à paciência da Intelligenesis em esperar pelos relatórios;
- ☺ a todos os outros com quem convivi — o pessoal da seção da pós, porteiros, faxineiras, o pessoal da lanchonete, da biblioteca, etc.;
- ☺ e à péssima programação da TV que me fez ficar mais tempo no laboratório.

ÍNDICE

Lista de Figuras	iii
Lista de Tabelas.....	vii
Resumo	vii
Abstract	viii
1 Introdução.....	1
1.1 CONTEXTUALIZAÇÃO.....	1
1.2 MOTIVAÇÃO E RELEVÂNCIA.....	5
1.3 OBJETIVOS	6
1.4 ORGANIZAÇÃO DA MONOGRAFIA.....	6
2 Etiquetadores de texto	7
2.1 ETIQUETAGEM MORFOSSINTÁTICA DE TEXTOS	8
2.1.1 <i>A importância da etiquetagem de textos</i>	9
2.2 A ETIQUETAGEM AUTOMÁTICA	9
3 Abordagens para Etiquetagem.....	14
3.1 ABORDAGEM LINGÜÍSTICA	15
3.1.1 <i>Etiquetador baseado em sintaxe</i>	17
3.1.2 <i>Etiquetador baseado em restrições</i>	18
3.1.3 <i>Etiquetador baseado em casos</i>	19
3.2 ABORDAGEM PROBABILÍSTICA.....	20
3.2.1 <i>Etiquetador Estatístico</i>	21
3.2.1.1 Xerox HMM.....	21
3.2.1.2 Um Etiquetador estatístico de categorias morfossintáticas para o português.....	22
3.2.1.3 TreeTagger.....	25
3.2.1.4 MXPOST	26
3.2.2 <i>Etiquetador Neural</i>	27
3.2.2.1 Net-Tagger	28
3.2.2.2 Etiquetador Neural da Universidade Nova de Lisboa.....	31
3.2.2.2 <i>Etiquetador Neural Elástico</i>	33
3.3 ABORDAGENS HÍBRIDAS	34
3.3.1 <i>Etiquetador baseado em transformação dirigida por erro</i>	36
4 Etiquetadores Treinados	41
4.1 DEFINIÇÕES DE PROJETO	41
4.1.1 <i>Corpus de treinamento e teste</i>	42
4.1.1.1 Etiquetagem manual de um corpus de treinamento e teste	44
4.1.2 <i>Abordagens de etiquetagem</i>	46
4.1.3 <i>Avaliação</i>	46
4.1.3.1 Avaliação da taxa de erro verdadeira	47

4.1.3.2 Fatores de impacto no processo de etiquetagem	51
4.2 RESULTADOS DOS EXPERIMENTOS COM OS ETIQUETADORES INDIVIDUAIS.....	52
4.2.1 <i>Avaliação dos métodos de etiquetagem</i>	54
4.2.1.1 TreeTagger	55
4.2.1.2 Etiquetador baseado em transformação (TBL).....	57
4.2.1.3 MXPOST	60
4.2.1.4 PoSiTagger – Portuguese Symbolic Tagger	62
4.2.2 <i>Avaliação do conjunto de etiquetas.....</i>	66
4.2.3 <i>Avaliação dos tipos de texto.....</i>	66
5 Combinação de etiquetadores.....	70
5.1 TAXA DE COMPLEMENTARIDADE DE ETIQUETADORES	74
5.2 MÉTODOS PARA COMBINAÇÃO DE ETIQUETADORES.....	75
5.2.1 <i>Métodos paralelos baseados em diferentes algoritmos com um único conjunto de dados de treinamento</i>	76
DECISÃO ALEATÓRIA	79
5.2.2 <i>Métodos paralelos baseados em um único etiquetador</i>	79
5.2.3 <i>Método em cascata aplicado ao TBL</i>	81
6 Discussão dos resultados.....	83
7 Conclusões e trabalhos futuros.....	87
7.1 CONTRIBUIÇÕES	87
7.2 TRABALHOS FUTUROS.....	88
Bibliografia e Referências	90
Apêndice A – NILC tagsets	101
A1 – NILC TAGSET (VERSÃO 1).....	103
A2 – NILC TAGSET (VERSÃO2).....	106
A3 – NILC TAGSET (VERSÃO 3)	107
A4 – NILC TAGSET (VERSÃO 4)	108
A5 – NILC TAGSET (VERSÃO 5 – VERSÃO).....	110
Apêndice B – Manual dos etiquetadores	112
B1 - TREETAGGER	112
B2 - TBL.....	115
B3 - MXPOST.....	117
B4 - POSITAGGER.....	118
Apêndice C – Regras para desambiguação gramatical.....	119
C1 - REGRAS PARA ETIQUETAGEM MORFOSSINTÁTICA DO REGRA	119
C2 - REGRAS DO ETIQUETADOR TBL	122
C3 REGRAS DO ETIQUETADOR X	134
Apêndice D – InCorpora	142
Glossário	150

LISTA DE FIGURAS

FIGURA 31 - PRECISÃO POR ETIQUETA DOS ETIQUETADORES TREETAGGER, TBL, MXPOST E POSITAGGER.....	66
FIGURA 32 - COMBINAÇÃO DE CLASSIFICADORES EM PARALELO.....	72
FIGURA 33 - COMBINAÇÃO DE CLASSIFICADORES EM CASCATA.....	72
FIGURA 34 - COMBINAÇÃO HIERÁRQUICA DE CLASSIFICADORES	72
FIGURA 35 - MÉTODOS PARA COMBINAR AS SAÍDAS EM UM MODELO PARALELO.....	76
FIGURA 36 - ALGORITMOS DE VOTAÇÃO SIMPLES E PONDERADA PARA A ETIQUETAGEM MORFOSSINTÁTICA.....	77
FIGURA 37 - REGRAS PARA PÓS-PREOCESAMENTO DE ÉNCLISES E MESÓCLISES.....	85
FIGURA 38 - FLUXO DE DADOS NO TREINAMENTO E ETIQUETAGEM COM O ETIQUETADOR TREETAGGER	114
FIGURA 39 - FLUXO DE DADOS NO TREINAMENTO COM O ETIQUETADOR TBL	117
FIGURA 40 - FLUXO DE DADOS NO TREINAMENTO E ETIQUETAGEM COM O ETIQUETADOR MXPOST	118
FIGURA 41 – FLUXO DE DADOS NO POSITAGGER.....	118

LISTA DE TABELAS

TABELA 1 - REGRAS DE CONTEXTO DO ETIQUETADOR BASEADO EM TRANSFORMAÇÃO DIRIGIDA POR ERRO	39
TABELA 2 - REGRAS PARA PALAVRAS DESCONHECIDAS.....	40
TABELA 3 - CORPUS DE TREINAMENTO E TESTE	44
TABELA 4 - TAXA DE ERRO NA ETIQUETAGEM MANUAL.....	45
TABELA 5 - COMPARAÇÃO ENTRE HOLDOUT E RANDOM SUBSAMPLING (BATISTA & MONARD, 1998).....	49
TABELA 6 - DIVISÕES DO CORPUS	52
TABELA 7 - TAXAS DE AMBIGÜIDADE NAS DIVISÕES DO CORPUS	53
TABELA 8 - PORCENTAGEM DE PALAVRAS DESCONHECIDAS.....	54
TABELA 9 - PRECISÃO GERAL DO ETIQUETADOR TREE TAGGER.....	55
TABELA 10 - TEMPOS DE TREINAMENTO E ETIQUETAGEM - TREE TAGGER.....	56
TABELA 11 - PRECISÃO POR ETIQUETA NO CORPUS DE TESTE - TREE TAGGER COMO TRIGRAMA	56
TABELA 12 - TEMPO DE TREINAMENTO E ETIQUETAGEM - TBL.....	58
TABELA 13 - PRECISÃO GERAL - TBL.....	58
TABELA 14 - PRECISÃO POR ETIQUETAS NO CORPUS DE TESTE- TBL	59
TABELA 15 - PRECISÃO GERAL - MXPOST	60
TABELA 16 - TEMPO DE TREINAMENTO E TESTE - MXPOST	60
TABELA 17 - PRECISÃO POR ETIQUETAS NO CORPUS DE TESTE - MXPOST	61
TABELA 18 - PRECISÃO GERAL DO POSITAGGER	64
TABELA 19 - TEMPOS DE ETIQUETAGEM DO POSITAGGER	64
TABELA 20 - PRECISÃO GERAL DO POSITAGGER NOS CORPUS DE TESTE E CALIBRAÇÃO.....	64
TABELA 21 - PRECISÃO GERAL DOS ETIQUETADORES PARA TEXTOS DIDÁTICOS.....	67
TABELA 22 - PRECISÃO GERAL DOS ETIQUETADORES PARA TEXTOS JORNALÍSTICOS.....	67
TABELA 23 - PRECISÃO GERAL DOS ETIQUETADORES PARA TEXTOS LITERÁRIOS	67
TABELA 24 - ETIQUETADORES DIDÁTICOS FRENTE A TEXTOS JORNALÍSTICOS E LITERÁRIOS.....	69
TABELA 25 - ETIQUETADORES JORNALÍSTICOS FRENTE A TEXTOS DIDÁTICOS E LITERÁRIOS.....	69
TABELA 26 - ETIQUETADORES LITERÁRIOS FRENTE A TEXTOS DIDÁTICOS E JORNALÍSTICOS.....	69
TABELA 27 - TAXA DE COMPLEMENTARIDADE ENTRE ETIQUETADORES	74
TABELA 28 - CONCORDÂNCIA ENTRE ETIQUETADORES NO TESTE.....	75
TABELA 29 - COMBINAÇÃO IDEAL	75
TABELA 30 - RESULTADOS DA COMBINAÇÃO.....	79
TABELA 31 RESULTADOS DA COMBINAÇÃO UTILIZANDO O MÉTODO TAGPAIR.....	79
TABELA 32 - PRECISÃO DA COMBINAÇÃO USANDO BAGGING	81
TABELA 33 - RESULTADOS OBTIDOS NOS MODELOS EM CASCATA.....	82
TABELA 34 - OPÇÕES DO COMANDO TAGGER.....	117

RESUMO

A etiquetagem morfossintática é uma tarefa básica, bem conhecida e bastante explorada em diversas aplicações de Processamento de Línguas Naturais (PLN), como análise sintática e extração e recuperação de informações. Os etiquetadores para a língua inglesa atingiram um estado da arte entre 96-99% de precisão geral. Diferentemente do inglês, para o português do Brasil não foram ainda exploradas todas as técnicas para a etiquetagem, nem se atingiu a precisão dos melhores etiquetadores para a língua inglesa. Com estas motivações, quatro etiquetadores disponíveis na WWW foram treinados — Unigrama (TreeTagger), Trigrama (TreeTagger), baseado em transformações (TBL) e baseado em máxima entropia (MXPOST) —, e um etiquetador simbólico foi desenvolvido (PoSiTagger). Todos os etiquetadores adaptados foram treinados com um corpus com cerca de 100.000 palavras formado por textos didáticos, jornalísticos e literários, e etiquetado com o Nilc tagset. A maior precisão geral obtida foi a do MXPOST — 89,66%. Foram também implementados quatorze métodos para a combinação dos etiquetadores, dos quais sete superaram a precisão do MXPOST. A maior precisão obtida com os métodos de combinação foi 90,91%. A precisão geral sofreu a influência do tamanho do corpus manualmente etiquetado disponível para treinamento, do conjunto de etiquetas e dos tipos de texto utilizados.

ABSTRACT

POS tagging is a very basic and well known natural language processing task used in several applications such as parsing and information retrieval. The taggers for English achieved a state of the art accuracy of 96-99%. Unlike the case of English, only some approaches to tagging were explored for Brazilian Portuguese and the tagging systems available are still unsatisfactory from the point of view of results based on the state-of-the-art accuracy for English. Four taggers have been trained with the NILC tagset on a mixed 100,000-word corpus of Brazilian Portuguese, namely Unigram (Treetagger), N-gram (Treetagger), transformation-based (TBL) and Maximum-Entropy tagging (MXPOST), and a symbolic tagger, named PoSiTagger, was designed. MXPOST displayed the best accuracy (89.66%). Fourteen methods of combination were used, seven of which led to an improvement over the MXPOST accuracy. The best result from the combination strategy was 90,91%. The low accuracy is attributed to the reduced size of the training corpus, the tagset used and the mixed corpus employed.

1 INTRODUÇÃO

The problems are difficult and numerous, and the possibility of achieving a high-quality output has been questioned. Why, then, do researchers persist in their efforts? In one sense this question is similar to asking why people struggle to climb mountains or to get to the moon. The mountain, or the moon, is a challenge, and science grows and makes new discoveries as it attempts to meet challenges. This is certainly true of the research efforts in automated language processing. – Harold Borko

1.1 Contextualização

Borko (1968) define processamento de língua natural como sendo a manipulação (codificação) de uma língua com propósitos específicos como comunicação, tradução, armazenamento e recuperação de informações. O processamento automático de línguas seria uma forma de facilitar a manipulação destas utilizando métodos computacionais. Imaginava-se que as tarefas que poderiam ser melhoradas através de sua execução automática seriam: a tradução de uma língua em outra, o armazenamento e recuperação de informações, e o projeto de um automôto inteligente capaz de responder perguntas.

Borko dizia que o processamento automático da língua nunca seria perfeito, mas que poderia ser de alta qualidade. E que apesar da língua ser indisciplinada, ingovernável, havia duas razões para que os pesquisadores continuassem desenvolvendo trabalhos em processamento automático de línguas:

- I) Uma razão relacionada a própria natureza humana – pesquisadores gostam de desafios, têm sede de novas descobertas.

- 2) Uma razão prática – caso não consigamos desenvolver métodos mais eficientes de comunicação para compartilhar idéias, o progresso humano seria inibido. Dada a importância da troca de informações entre pesquisadores, com a explosão de informações (que começou com a intensificação das pesquisas na guerra fria), gerou-se um grande volume de informações, surgindo a necessidade de se ter métodos que auxiliassem o pesquisador a ter acesso à literatura até mesmo para que não houvesse duplicação de trabalhos.

Nos últimos anos, com o avanço cada vez mais rápido dos computadores e da tecnologia relacionada, esta explosão de informações está cada vez mais intensa. Diariamente, trilhões de unidades de informação circulam pelo mundo. Graças a este avanço, textos em formato eletrônico são facilmente obtidos — textos clássicos de grandes escritores, textos de jornais, publicações científicas, etc. — com milhões de palavras e estruturas lingüísticas das mais variadas.

Esta grande variedade de textos fez com que o interesse por métodos empíricos de análise da língua ressurgisse entre o final da década de 80 e o início da década de 90, possibilitando o aumento do número de trabalhos na área de lingüística de *corpus*¹, visto que, para fins de análise lingüística é necessário o uso de textos representativos da língua, dialetos ou subconjuntos da língua. Surgiram, com este renascimento dos métodos empíricos e estatísticos, vários *corpora* grandes, como, por exemplo, o Birmingham Corpus (Sinclair et al., 1987), resultados de esforços como: Association for Computational Linguistics Data Collection Initiative (ACL/DCI), European Corpus Initiative (ECI), British National Corpus (BNC), Linguistic Data Consortium (LDC) Consortium for Lexical Research (CLR), Electronic Dictionary Research (EDR) e Text Encoding Initiative (TEI).

A pesquisa lingüística abrange uma variedade de níveis de análise. Um exemplo típico de ferramenta de análise seria o concordanceador. Um concordanceador é um programa que recupera todas as ocorrências de uma determinada cadeia de caracteres em um *corpus* e as listam, permitindo inclusive que estas listas sejam manipuladas. Contudo, certos tipos de análise não podem ser obtidos apenas através da grafia das palavras sem a utilização de outras características das palavras em questão, como por exemplo, informações de natureza gramatical.

Porém, para extrair o máximo de informações de um *corpus* através dessas ferramentas, é necessário fornecer, como entrada para elas, um *corpus* já etiquetado com marcas chamadas, no inglês, de *part-of-speech tags*. Tais marcas ou etiquetas são, principalmente, as categorias

gramaticais (morfossintáticas) das palavras do *corpus*. *Corpora* etiquetados são importantes para a construção de modelos estatísticos de gramáticas para a língua escrita formal e coloquial, desenvolvimento de teorias formais sobre as diferentes gramáticas da língua escrita e falada, investigação de fenômenos prosódicos na fala e avaliação de modelos de análise sintática. Existem vários esforços de pesquisas para marcar grandes *corpora* com informação lingüística, incluindo categorias gramaticais e estruturas sintáticas, por exemplo, o Penn Treebank (Marcus et al., 1993) e o British National Corpus (Leech et al., 1994).

As ferramentas utilizadas na etiquetagem automática de *corpus* são os etiquetadores (*taggers*). A etiquetagem automática é uma tarefa básica, bem conhecida e bastante explorada em Processamento de Línguas Naturais (PLN). É muito utilizada em diversas aplicações de PLN, como extração e recuperação de informações, por exemplo, na classificação de documentos em sistemas de busca da internet. Os etiquetadores para a língua inglesa atingiram um estado da arte entre 95-99% de precisão geral, visto que, independente da abordagem para etiquetagem escolhida alguns casos acabam não sendo tratados, por exemplo, por dependerem de informações semânticas, o que impõe um limite à precisão geral. Sempre haverá casos que não serão tratados, mesmo que tenhamos um lingüista genial para elaborar regras ou um etiquetador perfeito, isto porque não é possível construir um *corpus* que inclua todas as enunciação de uma língua dada ou subconjunto de uma língua, exceto para algumas línguas mortas, em que a quantidade de textos disponíveis é limitada.

Com a geração de léxicos a partir corpus de treinamento, a etiquetagem de textos do inglês vem se tornando cada vez mais fácil — cerca de 90% da precisão geral de um etiquetador é efeito do simples uso do léxico na etiquetagem (probabilidades léxicas) e, cerca de 50% dos 10% restantes pode ser resolvido por modelos simples de n-gramas. Mas ainda existem 50% dos 10% restantes que fazem com que a etiquetagem não seja uma tarefa trivial. Estes 50% são resultado, por exemplo, dos problemas:

- Ambigüidade léxica que não pode ser resolvida pelo contexto — existe a ambigüidade mas o contexto em que as palavras ambíguas aparecem é o mesmo. Por exemplo:

{Ele chegou rápido. (advérbio)
{Ele. chegou sério. (adjetivo)

¹ Os termos que estiverem sublinhados estão definidos no glossário.

- Tamanho de contexto inadequado — o contexto formado pelo período em que a palavra foco se encontra não é suficiente para resolver a ambigüidade. Por exemplo:

{ Quando o canto nada importa, a vida é triste. (canto é substantivo)
O hino nacional é fundamental. Quando o canto nada importa. (canto é verbo)

O primeiro período não constitui um problema para a etiquetagem, mas sim o terceiro, pois a ambigüidade só pode ser resolvida quando se conhece o segundo período.
- Palavras desconhecidas — palavras que não fazem parte do léxico, ou que não aparecem no léxico com a etiqueta correta para aquele contexto, ou que foram escritas de forma incorreta.
- Estruturas desconhecidas — estruturas que rompem com a estrutura comumente utilizada na formação de períodos e que não tenham aparecido no corpus de treinamento. Aparecem, por exemplo, em textos literários, naqueles que usam ordem indireta ou distanciam o objeto do verbo. Um exemplo que aparece em nosso corpus é o período:

{ Se a poder de estacas e diques o holandês extraiu_VT1 de um brejo salgado a Holanda, essa jóia do esforço, é que ali nada o favorecia.

Neste exemplo, todos os etiquetadores utilizados neste trabalho etiquetaram a palavra "extraiu" como verbo transitivo indireto, pois o objeto direto "a Holanda" veio depois do objeto indireto, o que usualmente não aconteceria em outros textos, como os textos jornalísticos. Já quando apresentamos o mesmo período reescrito em uma ordem mais usual — Se o holandês extraiu a Holanda, essa jóia do esforço, de um brejo salgado a poder de estacas e diques, é que ali nada o favorecia. — todos acertaram, etiquetando o verbo "extraiu" como verbo bitransitivo.
- Sentenças Labirinto — períodos para os quais não é possível uma análise morfossintática sequencial. Existem momentos em que se deve voltar e repensar a atribuição inicial das etiquetas, por exemplo:

{ Aluna precisa de matemática vence concurso.

Muito esforço foi feito na tentativa de se obter etiquetadores cada vez mais precisos para o inglês, como a etiquetagem manual de corpus volumoso, correção da etiquetagem automática também objetivando obter corpus de treinamento maior, desenvolvimento de novas técnicas supervisionadas e não supervisionadas e adaptação de técnicas utilizadas em Aprendizado de Máquina. O uso de técnicas de Aprendizado de Máquina se deve ao fato de etiquetadores poderem ser encarados como classificadores.

1.2 Motivação e Relevância

No Brasil, a pesquisa em PLN vem se intensificando há quase duas décadas. Vários sistemas para o português que trabalham em domínios limitados foram construídos no âmbito acadêmico (Beesley & Grefenstette, 1996; Bick, 1996; Nunes et al., 1996a, 1996b; Paiva et al., 1996). Porém, desde muito cedo se percebeu a complexidade do problema, oriunda tanto de problemas intrinsecamente lingüísticos, quanto das características próprias da língua portuguesa. Durante algum tempo, o entusiasmo pela área de PLN esmoreceu, ainda que vários resultados interessantes e promissores tenham sido alcançados. Atualmente, no entanto, a comunidade de PLN do Brasil vive um novo período de entusiasmo e resultados promissores. Um dos motivos para isso é a possibilidade de contar com ferramentas de tratamento lingüístico bastante abrangentes e independentes de língua, que facilitam sobremaneira a implementação de aplicativos nessa área. Os processadores de *corpus*, por exemplo, WordSmith², LEXA³ e MonoConc⁴ são ferramentas de muita utilidade, uma vez que são capazes de realizar importantes estatísticas sobre a língua escrita, subsidiando pesquisas sobre a língua e desenvolvimento de processadores automáticos de língua natural.

Dentro do convênio USP-Itautec, firmado em 1993, foi compilado um *corpora*⁵ de textos originais em português do Brasil, composto de textos jornalísticos, acadêmicos, literários, técnicos, empresariais, da constituição brasileira, etc. O *corpora* conta hoje com aproximadamente 35 milhões de palavras e não está etiquetado.

Diferentemente do inglês, para a língua portuguesa não foram ainda exploradas todas as técnicas para a representação de um modelo lingüístico da língua, nem se atingiu a precisão dos melhores etiquetadores para a língua inglesa. Existem três etiquetadores para o Português contemporâneo do Brasil — o etiquetador estatístico desenvolvido na UFRGS (Villavicencio, 1995), apresentando uma precisão de 84,5%, o etiquetador neural desenvolvido na Universidade Nova de Lisboa (Marques & Lopes, 1996), com a precisão 88.7%, e o etiquetador baseado em regras, desenvolvido por Eckhard Bick (Bick, 1996), que possui uma precisão acima de 99% –, e um etiquetador híbrido para o Português arcaico do Brasil (Finger, 1998; Alves, 1999; Alves & Finger, 1999; Galves & Britto, 1999).

² <http://www.liv.ac.uk/~ms2928/wordsmit.htm>

³ <ftp://www.hd.uib.no/pub/pc/lexa>

⁴ <ftp://ftp.nol.net/pub/users/athel/Win/monoconc/>

A etiquetagem automática para Português tem alguns outros pontos diferentes de trabalhos para inglês, como, por exemplo, o tamanho do corpus manualmente etiquetado disponível para treinamento (bem menor). Este e outros pontos serão tratados ao longo deste trabalho.

1.3 Objetivos

Este trabalho se propôs a construir um etiquetador simbólico, adaptar para o Português do Brasil três etiquetadores disponíveis via WWW (Schmid, 1995; Brill, 1994a; Ratnaparkhi, 1996), e combinar os etiquetadores adaptados utilizando técnicas da área de Aprendizado de Máquina. Pretendeu-se, assim, desenvolver um trabalho comparativo bastante extenso para a escolha de um etiquetador que etiquete com melhor precisão uma gama variada de tipos de texto em português do Brasil.

1.4 Organização da Monografia

A monografia está dividida em sete capítulos. O Capítulo 2 explica o que é a etiquetagem automática de textos. O Capítulo 3 trata das várias abordagens para etiquetagem encontradas na literatura e descreve as arquiteturas de alguns etiquetadores. O Capítulo 4 apresenta as decisões de projeto que foram tomadas neste trabalho com relação ao conjunto de etiquetas, corpus de treinamento e teste, abordagens de etiquetagem e métodos de avaliação dos etiquetadores e, traz também os experimentos realizados com cada um dos etiquetadores individualmente. O Capítulo 5 introduz a teoria de combinação de classificadores e traz os experimentos realizados com a combinação dos etiquetadores adaptados. O Capítulo 6 apresenta uma discussão dos experimentos realizados. Por fim, no Capítulo 7 estão as contribuições, limitações e sugestões de futuros experimentos e aprimoramentos.

⁵ <http://www.nilc.icmc.sc.usp.br/tools.html#CORPUS>

2 ETIQUETADORES DE TEXTO

Quando um rio corta, corta-se de vez
o discurso-rio de água que ele fazia;
cortado, a água quebra-se em pedaços,
em poços de água, em água paráltica.
Em situação de poço, a água equivale
a uma palavra em situação dicionária:
isolada, estanque no poço dela mesma,
e porque assim estanque, estancada;
e mais: porque assim estancada, muda,
e muda porque com nenhuma comunica,
porque cortou-se a sintaxe desse rio,
o fio de água por que ele discorría.

O discurso de um rio, seu discurso-rio
chega raramente a se reatar de vez;
um rio precisa de muito fio de água
para refazer o fio antigo que o fez.
Salvo a grandiloquência de uma cheia
lhe impondo interina outra linguagem,
um rio precisa de muita água em fios
para que todos os poços se enfrasem:
se reatando, de um para outro poço,
em frases curtas, então frase a frase,
até a sentença-rio do discurso único
em que se tem voz a sede ele combate.

João Cabral de Melo Neto

Este capítulo introduz o problema da etiquetagem automática de textos. Começa explicando o que é etiquetar morfossintaticamente um texto, e prossegue demonstrando a importância de se ter textos etiquetados para PLN e apresentando a estrutura de etiquetadores.

2.1 Etiquetagem morfossintática de textos

Marcar, anotar, ou etiquetar morfossintaticamente (*tagging*) um texto de uma dada língua significa atribuir um rótulo ou etiqueta (*tag*) de uma palette finita (*tagset*) a cada palavra da língua, símbolo de pontuação, palavra estrangeira, ou fórmula matemática de acordo com o contexto em que aparecem. Para as palavras da língua utiliza-se uma etiqueta referente a sua categoria morfossintática ou grammatical (adjetivo, verbo, substantivo, etc.); para os símbolos de pontuação aceitos no discurso (por exemplo, vírgula, ponto final, exclamação, interrogação, etc.) geralmente utiliza-se o próprio símbolo. As palavras estrangeiras e fórmulas geralmente são classificadas com um único rótulo, pois, embora, estejam na "borda" da gramática ou léxico da língua em que o texto é escrito, precisam ser etiquetadas (EAGLES, 1996b). Mostramos abaixo um esquema genérico de etiquetas para símbolos de um dado texto de uma língua.

1. N (substantivo)	5. AT (artigo)	9. NU (numeral)
2. V (verbo)	6. AV (advérbio)	10. I (interjeição)
3. AJ (adjetivo)	7. PE (preposição)	11. R (residual)
4. PO (pronome)	8. C (conjunção)	12. P (pontuação)

As etiquetas para cada classe grammatical das palavras de uma língua podem, por sua vez, ser refinadas com atributos referentes a cada classe. Por exemplo, para a língua portuguesa, os substantivos podem ser refinados com o valor para os atributos de tipo (comum, próprio), gênero (masculino, feminino, dois gêneros), número (singular, plural, dois números, invariável), grau (aumentativo, diminutivo, neutro). Pode-se, inclusive, chegar ao extremo de se criar uma etiqueta para uma palavra em particular; um exemplo disso é a etiqueta *VBR* do conjunto de etiquetas Claws7 (Wynne, 1996) que serve apenas para etiquetar a palavra *are*. Neste trabalho, estaremos usando o termo etiquetagem com o significado de etiquetagem morfossintática.

O processo de etiquetar consiste, então, em dada uma seqüência de símbolos do texto e um conjunto de etiquetas, associar a cada símbolo a sua respectiva etiqueta. O processo de etiquetagem é mostrado na Figura 1.

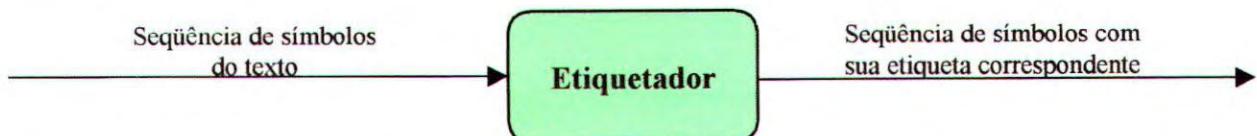


Figura 1 – Processo geral de etiquetagem de um texto

2.1.1 A importância da etiquetagem de textos

Estudos e análises de textos baseados em *corpora* utilizam geralmente *corpora* etiquetados pois conseguem assim extrair mais informação dos textos. Estes estudos utilizam grandes *corpora* (da ordem de milhões de palavras) para que as análises reflitam bem a realidade da língua, o que torna impraticável que os textos que estão sendo utilizados sejam etiquetados manualmente. Segundo Villavicencio (1995), a etiquetagem manual de um *corpus* com 20.982 de palavras feita por um linguista levou cerca de 44 horas, ou seja, para etiquetar manualmente um milhão de palavras nos mesmos moldes levaríamos cerca de 2097 horas. Assim, para etiquetar todo o *corpora* do Nilc, que possui 35.215.783 de palavras, usando o conjunto de etiquetas utilizado em (Villavicencio, 1995), levaríamos aproximadamente 73849 horas, ou seja, 8 anos 6 meses e 17 dias, isto sem contar férias, feriados, finais de semana, quinze minutos de atraso no expediente, etc. O alto custo da etiquetagem manual imprime a necessidade da etiquetagem automática.

A etiquetagem automática é uma técnica essencial em PLN. Pode ser aplicada em várias áreas do processamento de informação, incluindo: pré-processamento para summarização automática, pós-processamento para reconhecimento ótico de caracteres (OCR) e reconhecimento de fala, análise sintática (*parsing*), tradução automática e recuperação de informações.

2.2 A etiquetagem automática

Os programas que fazem a etiquetagem automática são chamados em inglês de *part-of-speech tagger* ou simplesmente *taggers*, língua para qual os primeiros deles surgiram. Na tradução para o

português são utilizados os termos etiquetador e rotulador. Neste trabalho será utilizado o termo etiquetador.

O processo de etiquetagem automática é composto basicamente por 3 tarefas que podem estar dispostas em 3 módulos: escrutinador léxico, classificador gramatical e desambigüizador gramatical, mostrados na Figura 2. No entanto, a maior parte dos etiquetadores pressupõe que os textos que lhe são fornecidos como entrada estão no formato inicial adequado e por isso não possuem um escrutinador léxico.

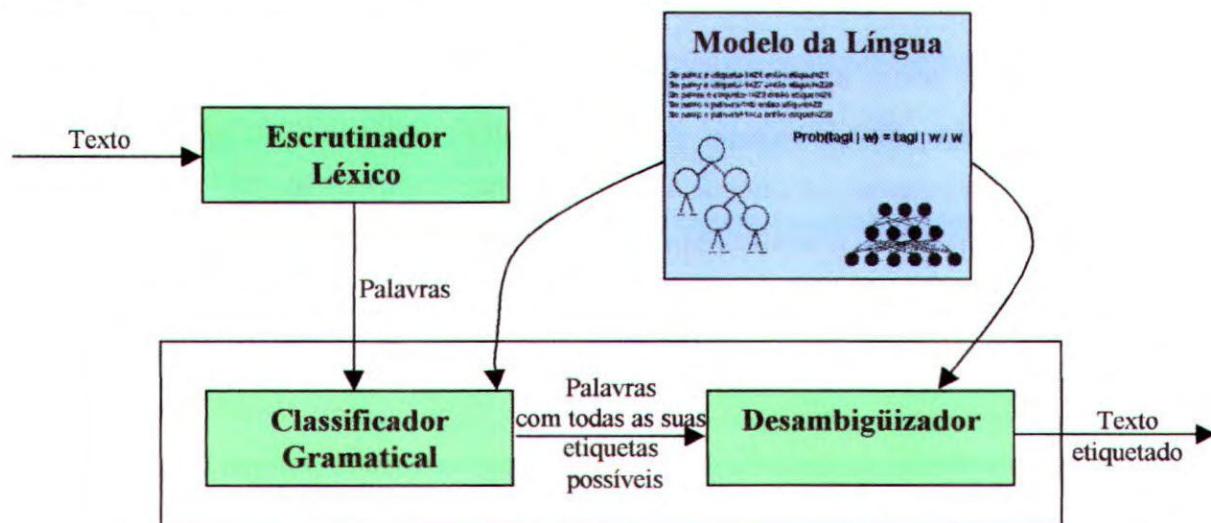


Figura 2 – Processo de etiquetagem automática

O escrutinador léxico identifica os símbolos do texto — pontuação, marcadores de documentos, palavras estrangeiras e as palavras da língua — separando dessa maneira os períodos do texto e, posteriormente, cada palavra de cada oração. A Figura 3 mostra o exemplo de um texto formado por dois períodos sendo submetido ao escrutinador lexical. No caso deste exemplo⁶, o escrutinador apenas acrescentou um espaço em branco antes do ponto final e colocou um período por linha.



⁶ O mesmo exemplo será utilizado nas Figuras 4 e 5 para ilustrar o funcionamento do classificador gramatical e do desambigüizador.

O classificador gramatical atribui classes gramaticais às palavras, utilizando, por exemplo, um léxico e um conjunto de informações para reconhecer as palavras que não pertencem ao léxico (ou por serem palavras estrangeiras, coloquialismos, regionalismos, ou simplesmente por não serem contempladas pelo léxico). Alguns exemplos de informações que podem ser utilizadas para este fim são: o primeiro/último/penúltimo/antepenúltimo caractere da palavra, se a palavra começa com letras maiúsculas, se a palavra tem outras letras maiúsculas além da primeira, se a palavra contém hífen, se contém algum caractere numérico. A Figura 4 ilustra qual seria a saída de um classificador gramatical que usasse o NILC tagset⁷ para o texto da Figura 3.

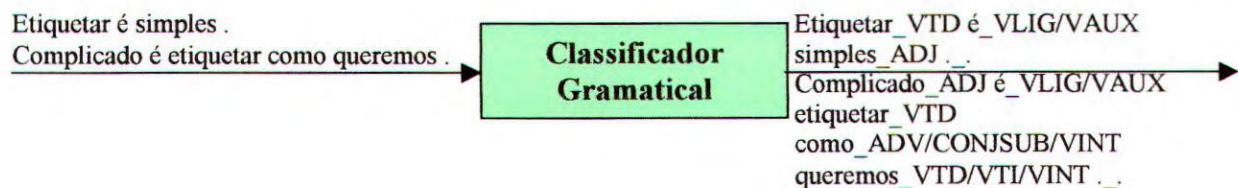


Figura 4 – Classificador Gramatical

Contudo, pode ocorrer o problema de ambigüidade lexical, situação em que temos palavras diferentes com mesma grafia, pertencendo a classes diferentes e que ocorre quando uma palavra é tratada isoladamente. Quando isso acontece, o contexto será analisado para solucionar o problema. Quem faz esta análise é o desambigüizador, que realiza a importante tarefa de atribuir uma, e somente uma, classe à cada palavra do corpus, como é visto no exemplo da Figura 5. A palavra *casa*, por exemplo, pode ser um substantivo ou um verbo. Essa ambigüidade pode ser resolvida quando se analisa o contexto em que ela se encontra. Por exemplo, na oração "Ela casa hoje", *casa* é um verbo, enquanto que na oração "Sua casa é bonita", *casa* é um substantivo.

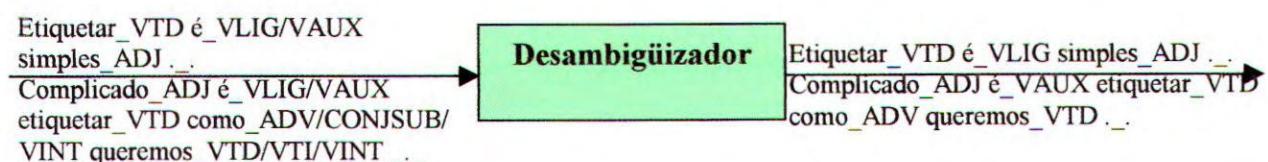


Figura 5 - Desambigüizador

⁷ Todas as versões do Nilc tagset estão presentes no Apêndice A deste trabalho. O manual referente à versão atual encontra-se disponível em (Disponível em <http://www.nilc.icmc.sc.usp.br/tools.html#TAGGER>

Tanto o léxico, quanto as informações utilizadas para avaliar o contexto fazem parte do modelo da língua utilizado por cada etiquetador. E é de acordo com a forma que o modelo da língua é construído e representado que classificamos os etiquetadores.

A primeira distinção entre os etiquetadores é feita de acordo com a maneira pela qual o modelo da língua é construído: manualmente, ou seja, elaborado por lingüistas ou automaticamente abstraído de *corpus*.

Os etiquetadores cujos modelos foram automaticamente abstraídos de um *corpus* de treinamento podem ainda ser divididos em supervisionados, não supervisionados e totalmente não supervisionados — segunda distinção. Os supervisionados são os etiquetadores que utilizam um conjunto de etiquetas e *corpus* de treinamento manualmente etiquetado como base para construção do modelo da língua. Já os não-supervisionados precisam apenas de um dicionário para saber qual é o conjunto de etiquetas e quais são as possíveis etiquetas para cada palavra, e de um pequeno *corpus* manualmente etiquetado para teste. Os totalmente não-supervisionados não dependem de um *corpus* manualmente etiquetado seja para treinamento, seja para teste, nem de um conjunto de etiquetas, pois fazem uso de métodos computacionais sofisticados para induzir agrupamentos de palavras (conjuntos de etiquetas).

Os etiquetadores supervisionados, não supervisionados e totalmente não-supervisionados podem ser subclassificados de acordo com a abordagem por eles utilizada para representar o modelo da língua: simbólicos (ou lingüísticos), probabilísticos e híbridos. Como exemplos de etiquetadores supervisionados simbólicos temos etiquetadores baseado em regras, baseado em casos e árvores de decisão não probabilísticas; como exemplos de probabilísticos temos os que fazem uso de Modelo de Markov, máxima entropia, árvores de decisão probabilísticas e redes neurais. De não supervisionados estatísticos temos os que usam o algoritmo Baum-Welch e o aprendizado baseado em transformação dirigida por erro (Brill, 1997b). E de totalmente não supervisionados temos os que usam análise distribucional de palavras de um texto, algoritmos de clusterização e redes neurais (Redington et al., 1998). A Figura 6 mostra o esquema da classificação dos etiquetadores. As abordagens para etiquetagem serão melhor explicadas no Capítulo 3.

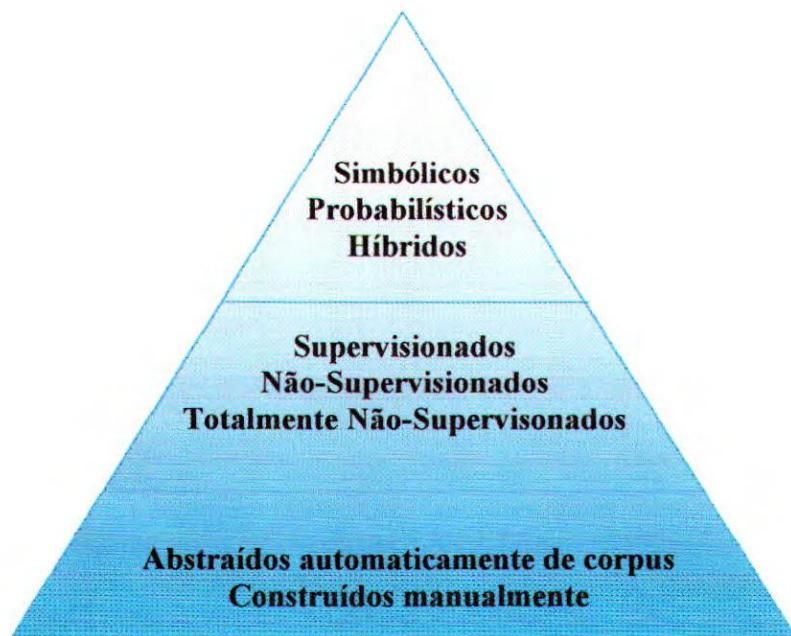


Figura 6 - Tipos de Etiquetadores

3 ABORDAGENS PARA ETIQUETAGEM

The best way to have a good idea is to have a lot of ideas. - Linus Pauling

Os primeiros etiquetadores seguiram a abordagem baseada em regras (Greene & Rubin, 1971, Klein & Simmons, 1963) e tiveram muito trabalho de construção manual; tinham léxicos pequenos que cuidavam basicamente de exceções às regras. O TAGGIT, de Greene & Rubin (1971), foi utilizado para fazer a etiquetagem inicial do Brown Corpus (Francis & Kucera, 1979), que foi depois manualmente revisado. Ele serviu de inspiração para a construção de um etiquetador estatístico, o CLAWS (Wynne, 1995, 1996).

Por volta de 1976, começaram a ser desenvolvidos etiquetadores estatísticos. A idéia de Hidden Markov Models (HMMs) até então utilizada no reconhecimento de fala começou a ser utilizada para etiquetagem automática de textos (Bahl & Mercer, 1976; Derouault & Merialdo, 1984; Church, 1988). Antes disso, pensava-se que a desambigüização morfossintática só teria um resultado realmente bom após a análise sintática ter sido realizada; não se acreditava que apenas o etiquetador fosse capaz de um bom trabalho de desambigüização. A partir dos primeiros etiquetadores estatísticos é que ficou comprovado que o etiquetador consegue obter uma boa taxa de acerto na categorização gramatical. No entanto, etiquetadores baseados em regras continuaram a ser desenvolvidos, só que buscando técnicas para ajudar na construção das regras (Karlsson, 1990; Voutilainen et al., 1992). Atualmente, os etiquetadores estatísticos são o tipo mais comum de etiquetadores. Basicamente, seu funcionamento consiste da construção de um

modelo estatístico da língua que é utilizado para desambigüizar uma sequência de palavras. Este modelo, em geral, aparece como um conjunto de frequências de diferentes tipos de fenômenos lingüísticos e é construído em geral através da observação de n-gramas, sendo que o mais comum é a modelagem na forma de unigramas, bigramas e trigramas.

Para melhorar a precisão dos etiquetadores, várias tentativas foram feitas. A partir de 1990 começaram a surgir estudos para a construção de etiquetadores híbridos, como por exemplo o de Brill (Brill, 1993b, 1994a, 1995, 1997b). Dentro da abordagem probabilística, surgiu a idéia do uso de árvores de decisão probabilísticas, de redes neurais artificiais (RN), e do modelo de máxima entropia. Surgiram também os etiquetadores não-supervisionados e totalmente não supervisionados que ajudam nos casos em que não há corpus manualmente etiquetado para uma dada língua ou domínio.

Constantemente, vários etiquetadores têm sido disponibilizados na WWW, tais como: HMM tagger (Cutting et al, 1992), Brill tagger (Brill 1994a, 1997b), TreeTagger (Schmid, 1995), MULTTEXT tagger (Armstrong et al., 1995), MXPOST (Ratnaparkhi, 1996) e TnT (Brants, 2000). Têm também sido desenvolvidos para várias línguas que não o inglês, por exemplo o francês (Chanod & Tapanainen, 1995), alemão (Feldweg, 1995; Schmid, 1995), grego (Dermatas & Kokkinakis, 1995), italiano (Dermatas & Kokkinakis, 1995), espanhol (León & Serrano, 1995; Márquez et al., 2000), sueco (Brants & Samuelsson, 1995), turco (Oflazer & Kuruöz, 1994), holandês (Dermatas & Kokkinakis, 1995), tailandês (Ma & Isahara, 1998; Ma et al., 1999a; Ma et al., 1999b; Lu et al., 2000), chinês (Ma et al., 1998), português (Villavicencio, 1995; Marques & Lopes, 1996; Bick, 1996; Finger, 1998; Alves & Finger, 1999; Galves & Britto, 1999).

Este capítulo discute de forma mais detalhada os etiquetadores simbólicos ou lingüísticos na Seção 3.1, os etiquetadores probabilísticos na Seção 3.2, e os híbridos na Seção 3.3, apresentando exemplos de etiquetadores supervisionados e não supervisionados.

3.1 Abordagem Lingüística

Um dos poucos resultados concretos obtidos nos primeiros vinte anos da pesquisa de I.A. é o fato de que a inteligência requer conhecimento. Para compensar sua característica predominante, a indispensabilidade, o conhecimento também possui algumas propriedades menos desejáveis, incluindo:

- É volumoso.
- É difícil de caracterizar com precisão.
- Está em constante mutação.

Elaine Rich

Na abordagem lingüística, são construídas regras a partir de abstrações lingüísticas sobre paradigmas e sintagmas da linguagem. Tais regras são codificadas manualmente como uma gramática que vai descartar análises ilegítimas. Devido ao fato da gramática ser construída manualmente, esta abordagem acaba exigindo habilidade lingüística, além de muito esforço para ser escrita. O processo de construção deste etiquetador é mostrado na Figura 7.

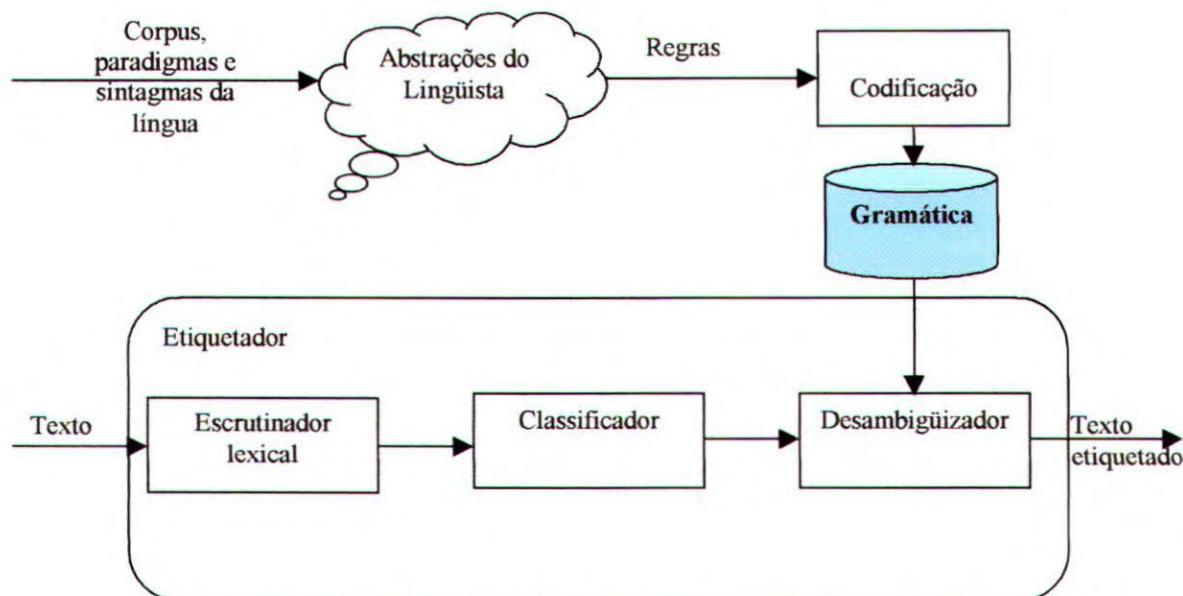


Figura 7 – Abordagem lingüística

A abordagem lingüística foi utilizada nas décadas de 70 e 80 para a análise sintática (Berwick et al., 1992) e baseia-se no pressuposto de que a categoria gramatical de uma palavra é uma característica arbitrária e convencional. Cada palavra é armazenada em um léxico, junto com sua(s) categoria(s) gramatical(ais), e outras informações relevantes para o processamento. Algumas características de um etiquetador lingüístico são:

- facilitar o processamento subsequente (análise sintática e interpretação semântica) com a riqueza de informações presente sobre cada item lexical;
- refletir o pensamento do lingüista que o construiu, já que este implementa as teorias que deseja usando regras;
- servir normalmente a um domínio específico, já que tanto as teorias quanto as generalizações da língua devem ser explicitamente definidas, portanto para um domínio maior seriam necessárias muitas regras (muito trabalho manual);
- utilizar normalmente programação simbólica.

Por melhor que seja o léxico sempre haverá falhas, porque o conjunto de palavras em determinado domínio é potencialmente infinito e na prática se modifica regularmente: há neologismos e empréstimos, palavras caem em desuso ou são modificadas (problema de incompletude). Alguns exemplos são os etiquetadores de Chanod & Tapanainen (1995a, 1995b), Voutilainen (1995), e Bick (1996).

3.1.1 **Etiquetador baseado em sintaxe**

Voutilainen (1995), apresenta um etiquetador que usa propriedades distribucionais gerais de um texto, que são invariantes através de línguas e sublínguas. Ele foi desenvolvido para o inglês e usou um *corpus* de treinamento com 38.000 palavras. Todas as regras gramaticais são essencialmente sintáticas. Consiste de cinco componentes seqüenciais: escrutinador, classificador, desambigüizador morfológico ENGCG, etiquetas sintáticas alternativas e desambigüizador sintático de estado finito, mostrados na Figura 8. Obteve 99,26% de precisão contra 95%-97% dos etiquetadores estatísticos devido à última fase, o desambigüizador sintático de estado finito.

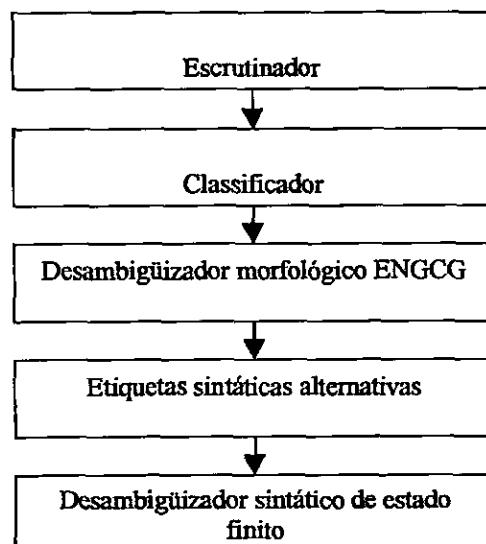


Figura 8 - Etiquetador baseado em sintaxe

No classificador, a descrição morfológica consiste de dois componentes: o léxico e regras heurísticas para tratamento de palavras desconhecidas. O léxico possui 80.000 entradas, cada uma

representando todas as formas flexionadas e algumas formas derivadas e emprega 139 etiquetas, principalmente para categorias gramaticais.

O classificador produz cerca de 180 classes de ambigüidade. As regras heurísticas têm análise baseada na formação das palavras e se nenhuma das regras é aplicável, a palavra é etiquetada como substantivo.

O desambigüizador morfológico ENGCG faz uso de uma Constraint Grammar. Esta gramática contém 1.185 restrições lingüísticas. São 200 as regras heurísticas alternativas (opcionais) e são baseadas em generalizações lingüísticas.

O desambigüizador sintático de estado finito consiste de uma gramática de Intersecção de estado-finito, um formalismo de análise sintática reducionista descrito detalhadamente em (Koskenniemi, 1990; Koskenniemi et al., 1992; Tapanainen, 1992; Voutilainen & Tapanainen, 1993; Voutilainen, 1994).

3.1.2 Etiquetador baseado em restrições

Chanod e Tapanainen (1995), fizeram um estudo para o francês que consistiu na construção de um etiquetador estatístico adaptado do Xerox HMM (Wilkins & Kupiec, 1996) e de um simbólico — o Etiquetador baseado em restrições — para o francês. Apesar de o estatístico ter precisão semelhante aos etiquetadores para o inglês — 96,8% de precisão — obteve-se melhor resultado com o simbólico, que possui 98,7% de precisão.

O Etiquetador baseado em restrições foi feito utilizando-se técnicas que foram originalmente desenvolvidas para análise morfológica (Karttunen, 1994). Neste modelo, as regras são representadas como analisadores de estado-finito. Os analisadores são compostos com períodos em seqüência. Cada analisador talvez remova ou mude uma ou mais etiquetas das palavras. Após todos os analisadores terem sido aplicados, cada palavra tem apenas uma análise.

A Figura 9 mostra o processo de etiquetagem utilizado. Para as formas de ambigüidade mais freqüentes são utilizadas restrições contextuais para resolver a ambigüidade. Através do estudo de um milhão de palavras retiradas de um jornal, Chanod e Tapanainen descobriram que o uso de restrições contextuais seria viável pois perceberam que as 97 palavras mais freqüentes

eram responsáveis por dois terços da ambigüidade e que dentre estas, apenas 16 (de, la, le, les, des, en, du, un, a, dans, une, pas, est, plus, Le, son) eram as responsáveis por 50% da ambigüidade⁸. Estas regras não requerem um corpus etiquetado e são independentes do corpus. São baseadas em uma pequena lista de palavras comuns (nas 97 mais freqüentes). Caso as regras principais não resolvam a ambigüidade são utilizadas heurísticas ad-hoc por exemplo: quando a palavra "des" aparece no início de frases ela é preferencialmente um determinador, por exemplo, descrições de combinações de nomes e frases preposicionais. Quando a regra anterior falha são utilizadas regras não contextuais que funcionam como probabilidades léxicas (por exemplo: a etiqueta preposição é preferível à etiqueta adjetivo, que é preferível à etiqueta pronome, que é preferível ao particípio passado).

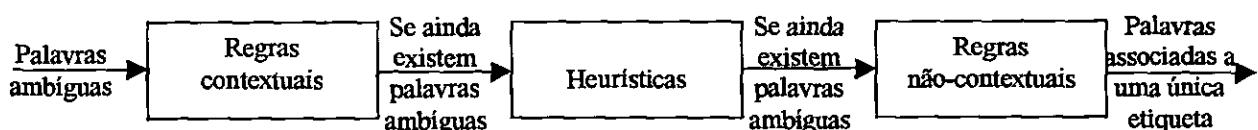


Figura 9 - Etiquetador baseado em restrições

Foram feitos dois tipos de teste, um com um corpus de 5752 palavras selecionadas de artigos sobre economia e outro com um corpus de 12000 palavras retiradas de textos de jornal, desta vez incluindo nomes próprios e siglas; a boa precisão se manteve nos dois tipos de texto. O conjunto de etiquetas utilizado possui 88 etiquetas. Contém 75 regras das quais: 39 são regras contextuais, 25 são regras heurísticas e 11 regras não-contextuais. Todas as regras são representadas por 11 analisadores de estado finito.

3.1.3 Etiquetador baseado em casos

O etiquetador MBT (Memory Based Tagger) foi proposto por (Daelemenans et al., 1996) e se baseia em raciocínio baseado em casos.

⁸ Um estudo semelhante foi feito para o inglês usando-se o Brown corpus. Descobriu-se que 63 palavras eram responsáveis por 50% da ambigüidade e que 220 palavras eram responsáveis por 2/3 da ambigüidade.

Durante a fase de treinamento, o corpus de treinamento é transformada em duas bases de casos, uma para ser usada nas palavras conhecidas e outra para palavras desconhecidas. Os casos são armazenadas em uma IGTree (memória de casos hierárquica (Daelemanset al., 1997)), e durante a etiquetagem novas casos são classificados combinando os novos casos com os casos que estão na memória indo do atributo mais importante para o menos importante. A ordem de relevância dos atributos é determinada utilizando Information Gain.

Para palavras conhecidas, o etiquetador tem informações sobre a palavra em foco e suas potenciais etiquetas, as etiquetas já desambigüizadas das duas palavras anteriores e as etiquetas ainda não desambigüizadas das duas palavras seguintes. Para palavras desconhecidas, são analisadas apenas uma palavra anterior e uma posterior, três letras de sufixo e atributos sobre a presença de letras maiúsculas, hífen e dígitos. A base de casos para palavras desconhecidas é formada apenas por palavras que apareceram no conjunto de treinamento cinco vezes ou mais.

3.2 Abordagem Probabilística

Há três tipos de situações em que é tentador utilizar o raciocínio probabilístico:
o mundo relevante é realmente aleatório;
o mundo relevante não é aleatório, dadas as informações suficientes,
mas nosso programa nem sempre terá acesso a todos os dados;
o mundo parece ser aleatório porque não o descrevemos ao nível certo.
Elaine Rich.

Baseia-se no pressuposto de que a categoria gramatical de uma palavra é o conjunto de seus modos de combinação com outras categorias (Harris, 1982). Isto é, em vez de ser uma característica intrínseca à palavra individual, a categoria gramatical representa um padrão de combinação de itens ou um componente de um padrão sintático.

Pela abordagem probabilística, portanto, não é necessário um conhecimento prévio dos itens individuais: suas possibilidades de combinação (probabilidade de transição de uma categoria para outra) são computadas diretamente do corpus. Se intensificou nos anos 80 e, tem se manifestado em sistemas que partem de levantamentos estatísticos detalhados em corpora e em sistemas de redes neurais.

3.2.1 Etiquetador Estatístico

Remember the commonsense idea that statistics is not a form of magic. Some sets of data contain too little information to answer our questions. Some questions cannot be answered at all. We cannot change this situation by indications or statistical formulas. But a grasp of statistical methods can increase our ability to learn from experience – sometimes in subtle ways. - Lincoln E. Moses

Na abordagem estatística não é necessário nenhum esforço humano, as abstrações (generalizações) são automaticamente adquiridas a partir do corpus. O único esforço necessário é o de determinar o conjunto de etiquetas adequado e marcar o corpus de treinamento. Porém, quando HMMs são utilizados há necessidade apenas de um léxico.

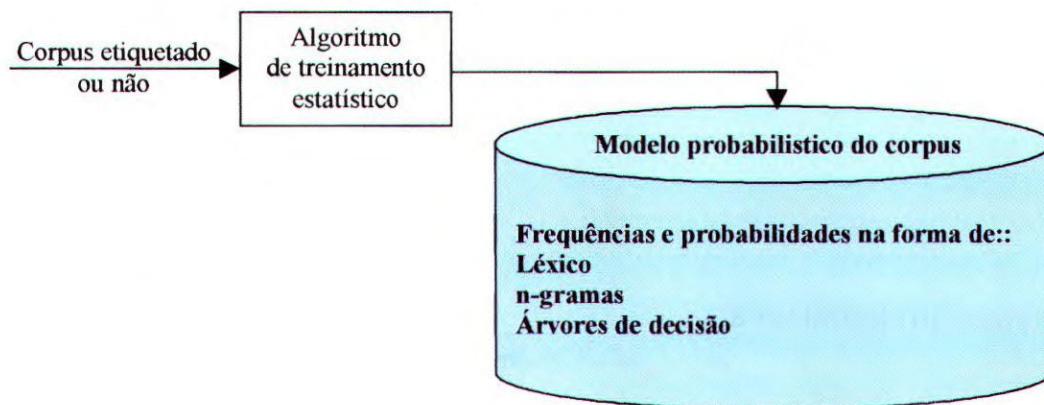


Figura 10 - Etiquetador Estatístico – Módulo de treinamento

Como o processo de inferir regras é totalmente automático, ao contrário da abordagem lingüística, aqui não existe um limite para o domínio, nem para a língua. Existem vários etiquetadores estatísticos que foram construídos a princípio para o inglês, tais como: Xerox HMM tagger (Wilkins & Kupiec, 1996) e TreeTagger (Schmid, 1995). Existe uma arquitetura que foi construída para o Português do Brasil (Villavicencio, 1995).

3.2.1.1 Xerox HMM

O Xerox HMM (Wilkins & Kupiec, 1996) é um etiquetador de domínio público baseado em HMM. Ele aplica o algoritmo Forward-Backward para treinamento e o algoritmo de Viterbi para a etiquetagem propriamente dita.

Para o treinamento, é necessário um escrutinador, um léxico, um corpus grande para treinamento (bons resultados foram obtidos com corpus a partir de 200.000 e os experimentos chegaram até corpus com 2.000.000 palavras) e um corpus manualmente etiquetado para teste. Pode ser treinado com texto etiquetado e/ ou não etiquetado. No entanto, os melhores resultados são obtidos quando se é utilizado no treinamento apenas corpora etiquetados.

O tamanho do conjunto de etiquetas varia de uma língua para outra, no caso da adaptação do Xerox HMM para o alemão, por exemplo, o conjunto continha 42 etiquetas (Feldweg, 1995). Sua precisão gira em torno de 96,67%.

3.2.1.2 Um Etiquetador estatístico de categorias morfossintáticas para o português

Trata-se de um etiquetador probabilístico simples que não contém nenhuma espécie de mecanismo para recalcular probabilidades. Foi desenvolvido pelo grupo de PLN da Universidade Federal do Rio Grande do Sul em conjunto com a Universidade Nova de Lisboa (Portugal) (Villavicencio, 1995) e apesar de ter sido desenvolvido para o português, nada impede que seja usado para outras línguas caso se tenha disponível um corpus para o treinamento.

Foram utilizados dois corpus para o treinamento: o Lusa Corpus (português Continental) e o Radiobrás (português do Brasil). O corpus Radiobrás possui 141.043 palavras, das quais 20.982 foram manualmente etiquetadas. O conjunto de etiquetas utilizado para a língua portuguesa do Brasil contém 33 etiquetas e para a de Portugal 45. Das 20.982 palavras, 20.000 foram utilizadas para treinamento e 982 para teste, obtendo 74,54% de precisão com dicionário aberto e corpus acentuado e 87,98% de precisão com dicionário fechado e corpus acentuado.

O etiquetador realiza a análise de um corpus de treinamento e modela os padrões lingüísticos nele presentes. Após ter modelado este conhecimento, o etiquetador pode usá-lo para fazer a etiquetagem automática das palavras de um outro corpus qualquer.

É composto de três módulos principais: o módulo construtor de HMMs, o módulo classificador e o módulo de Viterbi como mostra a Figura 11.

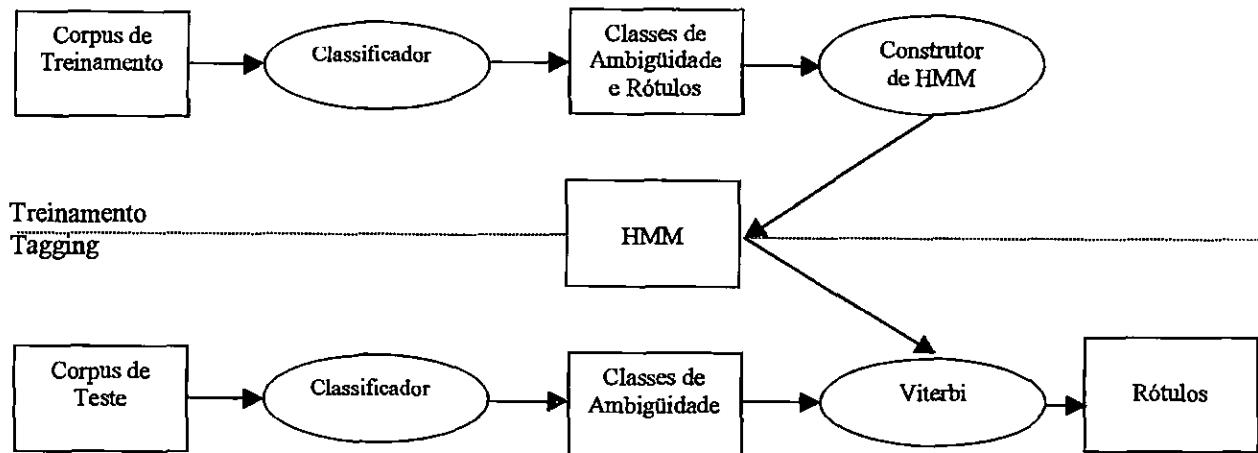


Figura 11 – Arquitetura do etiquetador estatístico de categorias morfossintáticas para o português (Villavicencio, 1995)

No módulo classificador, um corpus de entrada é lido e cada período é decomposto em palavras (Figura 12). Para cada palavra é atribuída uma classe de ambigüidade segundo definido anteriormente no dicionário.

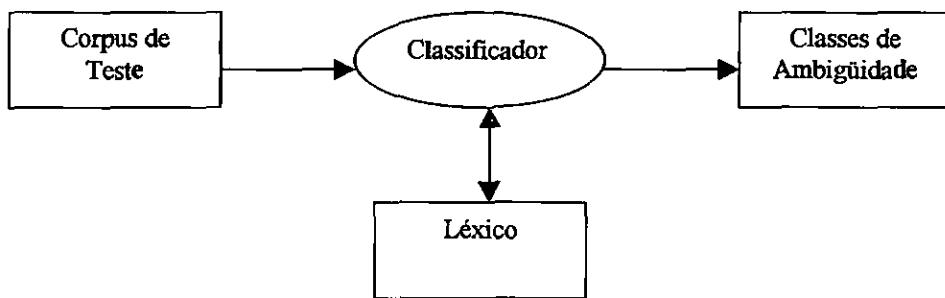


Figura 12 –Módulo classificador do etiquetador estatístico de categorias morfossintáticas para o português (Villavicencio, 1995)

Em seguida, no módulo construtor de HMMs baseado na análise dos padrões lingüísticos que ocorrem no corpus de treinamento, o construtor produz um HMM (Figura 13). Para modelar as probabilidades contextuais é utilizado o modelo de bigramas e para o cálculo das estimativas destas probabilidades, é utilizado o algoritmo de freqüência relativa. Após a modelagem das probabilidades é feito um refinamento, para evitar que se use probabilidades que tenham sido estimadas a partir de freqüências muito baixas. Para o refinamento é utilizado o algoritmo Deleted

Interpolation (para maiores detalhes veja (Jelinek & Mercer, 1980)). Após o modelo ter sido refinado, ele está pronto para utilizar o conhecimento adquirido para marcar outros corpus.

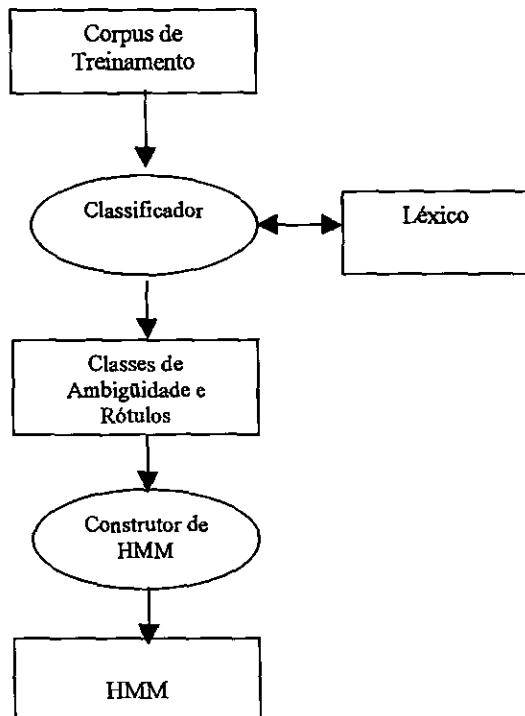


Figura 13 - Módulo Construtor do etiquetador estatístico de categorias morfossintáticas para o português (Villavicencio, 1995)

O último módulo, responsável pela resolução de ambigüidades, é o módulo de Viterbi. Este módulo analisa as classes de ambigüidade encontradas em um corpus (Figura 14). Através da aplicação do algoritmo de Viterbi sobre o HMM treinado, ele descobre qual a seqüência de rótulos mais provável.

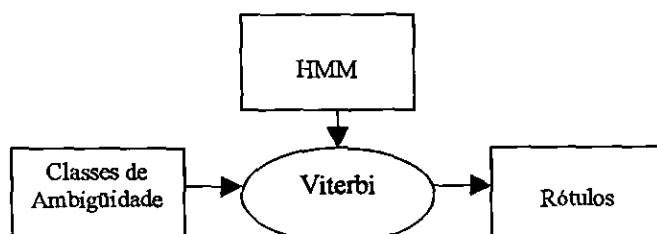


Figura 14 - Módulo de Viterbi do etiquetador estatístico de categorias morfossintáticas para o português (Villavicencio, 1995)

3.2.1.3 TreeTagger

O TreeTagger é um etiquetador baseado em árvore de decisão e modelo de Markov de segunda ordem. Apenas corpora etiquetados são usados para treinamento. Árvores de decisão são utilizadas para diminuir o problema de dados esparsos, obtendo estimativas confiáveis das probabilidades de transição. É que devido ao grande número de parâmetros (particularmente no caso de trigramas), os métodos estatísticos, em geral, encontram dificuldades na estimativa de probabilidades muito pequenas quando é utilizado um conjunto limitado de dados de treinamento. Essa árvore de decisão determina automaticamente o tamanho apropriado do contexto utilizado para estimar essas probabilidades (Figura 15).

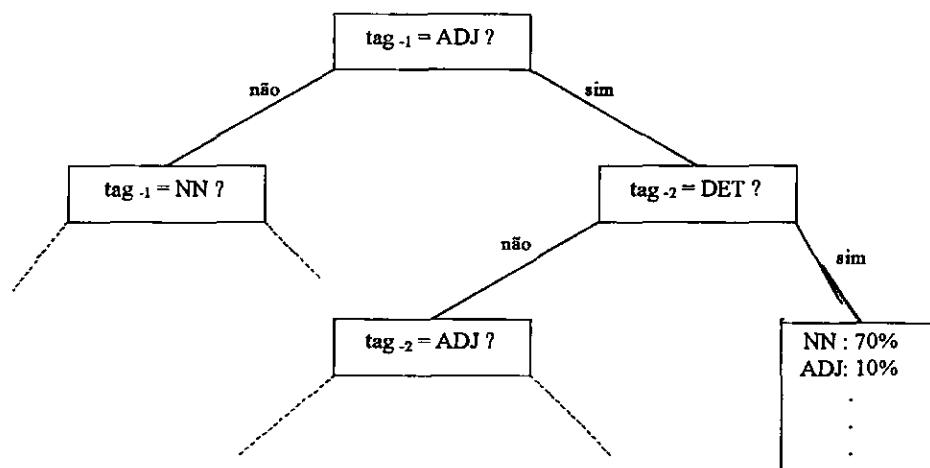


Figura 15 – TreeTagger - Árvore de decisão (Schmid, 1995)

A probabilidade de um dado trígrama é determinada seguindo o caminho correspondente através da árvore de decisão até que um nó folha seja encontrado.

As etiquetas para palavras desconhecidas são descobertas através de um léxico com sufixos que é construído automaticamente. Para probabilidades léxicas são utilizadas classes de ambigüidade.

Para avaliar o desempenho do TreeTagger para a língua inglesa foi utilizando o corpus Penn Treebank (Marcus et al., 1993), que utiliza 36 etiquetas e outras 12 para caracteres de pontuação. Aproximadamente 2 milhões de palavras foram usadas para treinamento e 100 mil palavras de uma parte diferente do corpus foi usada para teste.

O TreeTagger foi comparado, com os mesmos dados, com um etiquetador comum baseado em trigramas (Kempe, 1993) que não utiliza árvore de sufixos. Como resultado, obteve-se uma precisão de 96,36% por parte do TreeTagger, enquanto que o etiquetador estatístico comum alcançou uma precisão de 96,06%. Pode parecer uma diferença muito pequena, mas é considerável quando se trata de um grande volume de dados.

Foi também avaliada a influência do tamanho do corpus de treinamento na precisão do etiquetador. Ao contrário do etiquetador baseado em trigramas, a precisão do TreeTagger deteriora vagarosamente à medida que o corpus de treinamento diminui. Isto mostra que o TreeTagger é robusto com relação ao tamanho do corpus de treinamento em contraste com outros etiquetadores estatísticos.

3.2.1.4 MXPOST

MXPOST é um etiquetador baseado no modelo de Máxima Entropia e foi desenvolvido por Ratnaparkhi (1996). Utiliza várias características da palavra e do contexto de forma similar ao etiquetador MBT.

O modelo final tem um parâmetro de peso para cada valor de atributo que for relevante para estimar a probabilidade $P(\text{etiqueta} | \text{atributos})$, e combina as indicações de diversos atributos em um modelo probabilístico explícito. Em contraste com outros etiquetadores, as palavras conhecidas e desconhecidas são processadas pelo mesmo método.

Este etiquetador também não possui um mecanismo para armazenar em separado informações léxicas sobre a palavra em foco. A palavra é apenas mais um atributo no modelo probabilístico. Como resultado disto, não é possível fazer generalizações sobre grupos de palavras para as quais o mesmo conjunto de etiquetas é possível.

Na fase de etiquetagem, o algoritmo de busca beam search é utilizado para encontrar a sequência de etiquetas mais provável para todo o período.

3.2.2 Etiquetador Neural

As pessoas aprendem mudando a estrutura da rede neural que compõe seus cérebros. Por que então não deveríamos construir um programa de aprendizado começando com uma rede simples e fazer conexões nessa rede conforme indicado pelo esforço no ambiente?
Elaine Rich

Etiquetadores neurais fazem o “levantamento” do corpus implicitamente, como parte de seu funcionamento (Benello et al., 1989).

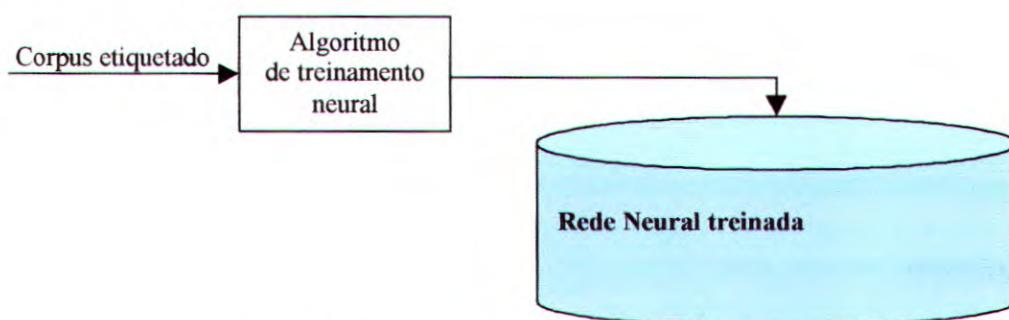


Figura 16 - Etiquetador Neural – Módulo de treinamento

Tem como vantagens as mesmas dos estatísticos: dispensa a elaboração prévia do léxico que é elaborado automaticamente, e é altamente adaptável. No entanto, também apresenta problemas. A fase de treinamento de redes neurais exige cuidado na elaboração de esquemas apropriados de treinamento para evitar superespecialização ou supergeneralização das redes (Benello et al., 1989; Braga et al., 1998).

Um outro fator interessante é que o etiquetador neural não precisa de um corpus etiquetado muito grande, o que facilita no caso da etiquetagem do português já que não existem corpus para o português com 1 milhão de palavras como existem para o inglês. Baseado nisto, na Universidade Nova de Lisboa, em Portugal, foi desenvolvido um etiquetador neural que foi treinado para o português continental e português do Brasil (Marques & Lopes, 1996). Um outro exemplo clássico é o Net-Tagger (Schmid, 1994).

3.2.2.1 Net-Tagger

O Net-Tagger é um etiquetador neural que foi desenvolvido para o inglês por Schmid (1994). Consiste de uma rede neural MLP (Multilayer Perceptron) com uma camada simples feed-forward para a desambigüização e de um léxico (Figura 17). A entrada da rede consiste de vetores de probabilidades léxicas das etiquetas para as palavras na contexto da palavra corrente. A saída da rede é também um vetor de probabilidades que alimenta a rede como vetor de probabilidade da palavra anterior. O etiquetador é treinado com corpus etiquetado usando uma modificação do algoritmo backpropagation. Assim como o TreeTagger, o Net-Tagger também usa um léxico de sufixos para descobrir as possibilidades de etiquetas para uma palavra desconhecida.

Para seu treinamento usou-se 2.000.000 palavras do Penn Treebank Corpus. Para teste utilizou-se um corpus com 100.000 palavras. O contexto limitou-se as três palavras anteriores ($p = 3$) e as duas seguintes ($f = 2$).

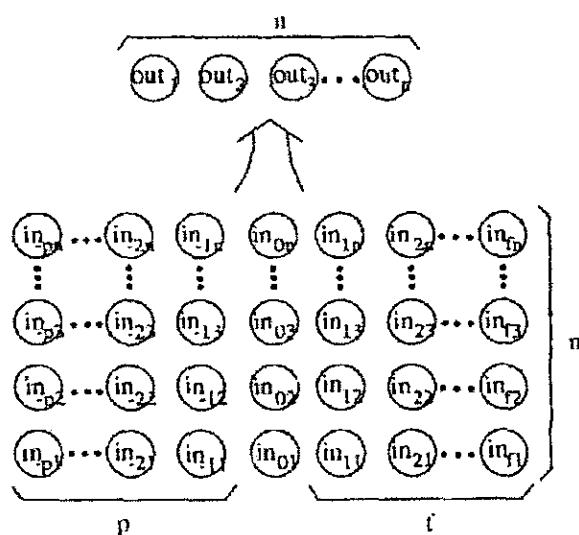


Figura 17 - A arquitetura do Net –Tagger (Schmid, 1994)

A rede neural gastou 4 milhões de ciclos no treinamento. O treinamento foi feito em uma Sparc 10 e levou um dia. Para a etiquetagem de um corpus com 100.000 palavras levou 12 minutos. O Net-Tagger obteve 96.22% de precisão, contra 96,06% do trigram tagger (Kempe, 1993) e 94,44% do HMM tagger, usando os mesmos dados de treinamento e teste.

3.2.2.1.1 A Rede Neural do Net-Tagger

Na camada de saída da rede MLP, cada unidade corresponde a uma das etiquetas do conjunto de etiquetas. A rede aprende durante o treinamento a ativar a unidade de saída que representa a etiqueta correta e a desativar as demais unidades. Portanto, na rede treinada a unidade de saída com maior ativação indica qual etiqueta deve ser anexada à palavra que está sendo etiquetada naquele instante.

A entrada da rede contém toda a informação que o sistema tem sobre as possíveis classificações da palavra corrente, das p palavras que a antecedem e das f seguintes. Ou seja, para cada etiqueta pos_j e cada uma das p + l + f palavras no contexto, há uma unidade de entrada cuja ativação in_{ij} representa a probabilidade da palavra_i ter a etiqueta pos_j.

Da palavra que está sendo etiquetada e das palavras seguintes, a única informação conhecida é a probabilidade léxica P(pos_j|palavra_i), que nada mais é do que o número de vezes que a palavra aparece com esta etiqueta dividido pelo número de ocorrências da palavra (Maximum Likelihood Principle). Como esta probabilidade, não leva em conta nenhuma influência contextual, a palavra que está sendo etiquetada e as palavras seguintes possuem a mesma representação de entrada (1):

$$in_{ij} = P(pos_j | palavra_i), \text{ se } i \geq 0 \quad (1)$$

Já sobre as palavras anteriores se tem mais informação, visto que já faram etiquetadas. Então, ao invés de se utilizar a probabilidade léxica, é utilizado o valor de ativação de saída da unidade do instante do tempo de processamento (2):

$$In_{ij}(t) = out_j(t+i), \text{ se } i < 0 \quad (2)$$

Copiar a ativação de saída da rede na entrada introduz recorrência dentro da rede. Isto complica o processo de treinamento, porque a saída da rede não está correta quando o treinamento é iniciado. Portanto, ao invés disso, é utilizada a "maior média da saída atual" no início do treinamento.

A rede é treinada com um corpus etiquetado. Os valores de ativação são zero para todas as unidades de saída, exceto para a unidade que corresponde a etiqueta correta que recebe o valor de ativação 1.

A rede pode usar ou não camadas intermediárias. Com camadas intermediárias o resultado obtido pode ser melhor. No entanto, uma rede com camadas intermediárias precisa de mais treinamento do que uma sem e também corre o risco de ter problemas com a capacidade de generalização (quando a rede começa a aprender coisas irrelevantes e se torna incapaz de generalizar).

Nos dois tipos de rede, a etiquetagem de uma palavra é feita copiando as etiquetas prováveis da palavra corrente e suas vizinhas nas unidades de entrada, propagando as ativações pela rede até as unidades de saída e vendo qual unidade de saída possui o maior valor de ativação. A etiqueta correspondente a esta unidade de maior valor de ativação será anexada a palavra atual. Se o segundo maior valor de ativação for muito próximo do primeiro, podemos ter uma etiqueta alternativa.

3.2.2.1.2 O Léxico do Net-Tagger

O léxico contém as etiquetas prováveis para cada palavra. Tem três partes: *fullform*, *suffix* e *default*.

Quando se vai procurar uma palavra no léxico, a busca é feita primeiro no léxico *fullform*. Se a palavra for encontrada, é retornado um vetor de etiquetas prováveis, se não, letras maiúsculas são transformadas em minúsculas e a busca é feita novamente. Caso após a segunda busca a palavra não tenha sido encontrada, a busca é feita então no léxico *suffix*. Se nenhum destes passos tiver resolvido, a entrada *default* do léxico é retornada.

O léxico *fullform* é criado a partir do corpus de treinamento. Depois, o número de ocorrências de cada par palavra/etiqueta é contado. Em seguida, todas as etiquetas com probabilidade estimada em menos de 1% são removidas, isto porque em grande parte dos casos elas são o resultado de erros no corpus original.

A segunda parte do léxico, o léxico *suffix*, forma uma árvore. Cada nó da árvore, exceto a raiz é marcado com um caracter. No final, vetores de etiquetas prováveis são anexados. Durante a busca, a árvore de sufixos é olhada a partir da raiz. A busca se inicia a partir do fim da palavra.

Por exemplo, a palavra *tagging* é procurada no léxico *suffix* mostrado na Figura 18. A busca se inicia no nó raiz e vai para o nó com o caracter *g*, em seguida se move para o nó *n* e logo depois para o nó *i*. O vetor de prováveis etiquetas é então retornado.

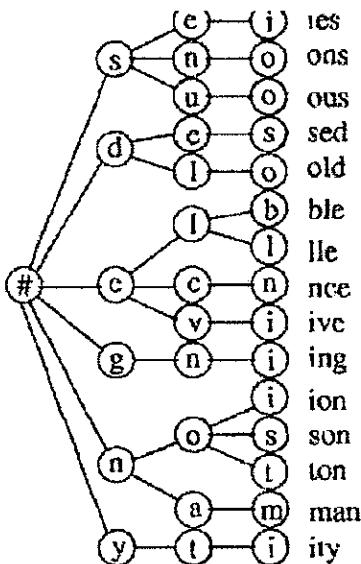


Figura 18 - Léxico Suffix do Net-Tagger (Schmid, 1994)

O léxico *suffix* é construído automaticamente a partir do corpus de treinamento. Primeiro, a árvore de sufixos é construída a partir de todas as palavras que tenham pelo menos comprimento cinco e que tenham sido etiquetadas como de classe aberta e então a freqüência das etiquetas é contada para cada um destes sufixos e adicionada a cada nó correspondente.

3.2.2.2 Etiquetador Neural da Universidade Nova de Lisboa

Este etiquetador neural foi desenvolvido por Marques e Lopes (1996) na Universidade Nova de Lisboa em Portugal. Ele será referenciado como Etiquetador de Portugal.

Ao contrário do Net-Tagger, que utilizou um código para a rede neural, as topologias de redes neurais foram testadas e treinadas usando um simulador de redes neurais: o Stuttgart Neural Network Simulator (SNNS). As unidades neurais são de três tipos: entrada, intermediária e saída. O

SNNS possui vários algoritmos de aprendizado, mas os únicos utilizados neste trabalho foram o backpropagation e o momentum backpropagation.

Os valores de entrada são recebidos diretamente de um arquivo chamado pattern. Os valores de saída podem ser passados para um arquivo (quando se está utilizando a rede neural para marcar um texto), ou podem ser recebidos do arquivo pattern (quando se está treinando a rede). O arquivo pattern pode ter vários vetores de entrada com valores de saída associados. Pode-se também ordenar os vetores de entrada em conjuntos com os valores de saída sendo computados para os conjuntos. O trabalho com conjuntos de vetores serve para possibilitar o uso do modelo de n-gramas. As unidades intermediárias podem também receber os valores das unidades anteriores e passar os para as próximas.

Cada unidade no SNNS tem duas funções: função de ativação e função de saída. No trabalho de Portugal todas as redes fizeram uso da função de ativação logística.

Após o corpus ter sido processado, o SNNS pode ser utilizado para marcar o texto. Estes procedimentos foram implementados usando comandos de processamento de texto do Unix, a linguagem de programação awk e uma ferramenta desenvolvida previamente chamada classifier. As ferramentas desenvolvidas no UNIX foram:

- classifier_795 – responsável por associar o vetor de probabilidades com cada palavra do corpus é usualmente chamado pelos scripts: evaluate_tagger.x e train_tagger.x;
- build_dic_795.x – constrói o dicionário usando o corpus etiquetado como base;
- rand_split.x – este script divide o corpus randomicamente;
- create_tdnn – transforma arquivos de treinamento ou de teste em arquivos no formato requerido pelo SNNS;
- evaluate_tagger.x – marca o corpus de teste com a rede usada anteriormente e compara o resultado com o corpus etiquetado manualmente, calculando o acerto;
- train_tagger.x – prepara o arquivo de treinamento para treinar a rede;
- unigram_tagger.x – Dá a base-line, através da associação de cada palavra do corpus à etiqueta mais freqüente para aquela palavra.

conhecimentos inconsistentes no sistema. Mas, apesar de todos estes problemas, a precisão dos etiquetadores estatísticos é bastante alta: 95-97 %, principalmente para o inglês (Voutilainen, 1995).

Com todos estes aspectos, o que se percebe é que se deve pensar cuidadosamente a respeito da abordagem que se vai utilizar. Ou, então, optar por uma abordagem híbrida, que é uma alternativa bastante interessante porque aumenta as vantagens de cada uma em particular e diminui as desvantagens encontradas na abordagem probabilística e lingüística.

Eric Brill (1994a), descreve um etiquetador contendo um componente estatístico e outro simbólico. O componente estatístico dá sua contribuição somente durante o treinamento, produzindo regras que são puramente simbólicas. Quando usado põe em ação estas regras que podem ser modificadas. Já o trabalho de Pacheco (Pacheco et al., 1996) é simbólico ao tratar dos itens funcionais, e é probabilístico ao tratar dos itens de classes abertas. No entanto, é feita uma avaliação sobre o que realmente dentre estes itens de classes abertas necessita de uma abordagem probabilística para que não haja desperdício de tempo de processamento. Nesta abordagem, a etiquetagem gramatical tenta utilizar ao máximo o potencial das abordagens estatística e simbólica. Baseia-se no fato de que um texto de uma dada língua possui algumas características básicas bem definidas que se repetem freqüentemente nos textos. Quando se faz uso apenas da abordagem probabilística acaba-se desperdiçando recursos e tempo de processamento. A solução dada a este problema foi aproveitar da abordagem simbólica a utilização de informações preestabelecidas somente para os itens funcionais, que aparecem com muita freqüência em textos para todos os assuntos. E da probabilística aproveitar a utilização de procedimentos dependentes de contexto para o estabelecimento da etiquetagem sintática dos elementos das classes gramaticais abertas, que variarão de um contexto para outro e de assunto para assunto. Um etiquetador sob esta abordagem possuiria um estado inicial simbólico e um final estatístico (Figura 20), exatamente o inverso do etiquetador de Brill.

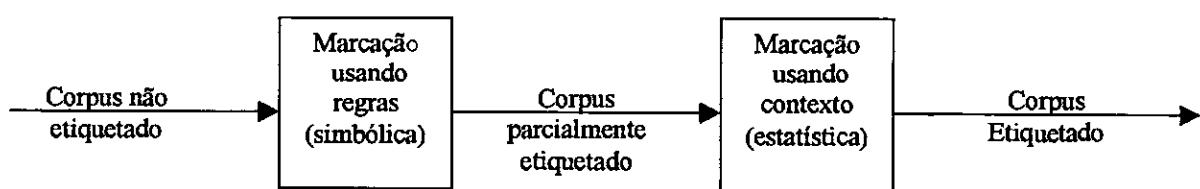


Figura 20 - Etiquetador segundo a abordagem funcionalista

3.3.1 Etiquetador baseado em transformação dirigida por erro

Este etiquetador híbrido baseia-se no algoritmo de Aprendizado Baseado em Transformação Dirigida por Erro (Brill, 1994a, 1995, 1997b) desenvolvido por Brill (Figura 21). Tal algoritmo pode ser aplicado a vários problemas, como, por exemplo, a etiquetagem morfossintática e a análise sintática (Brill, 1994a, 1996a, 1996b).

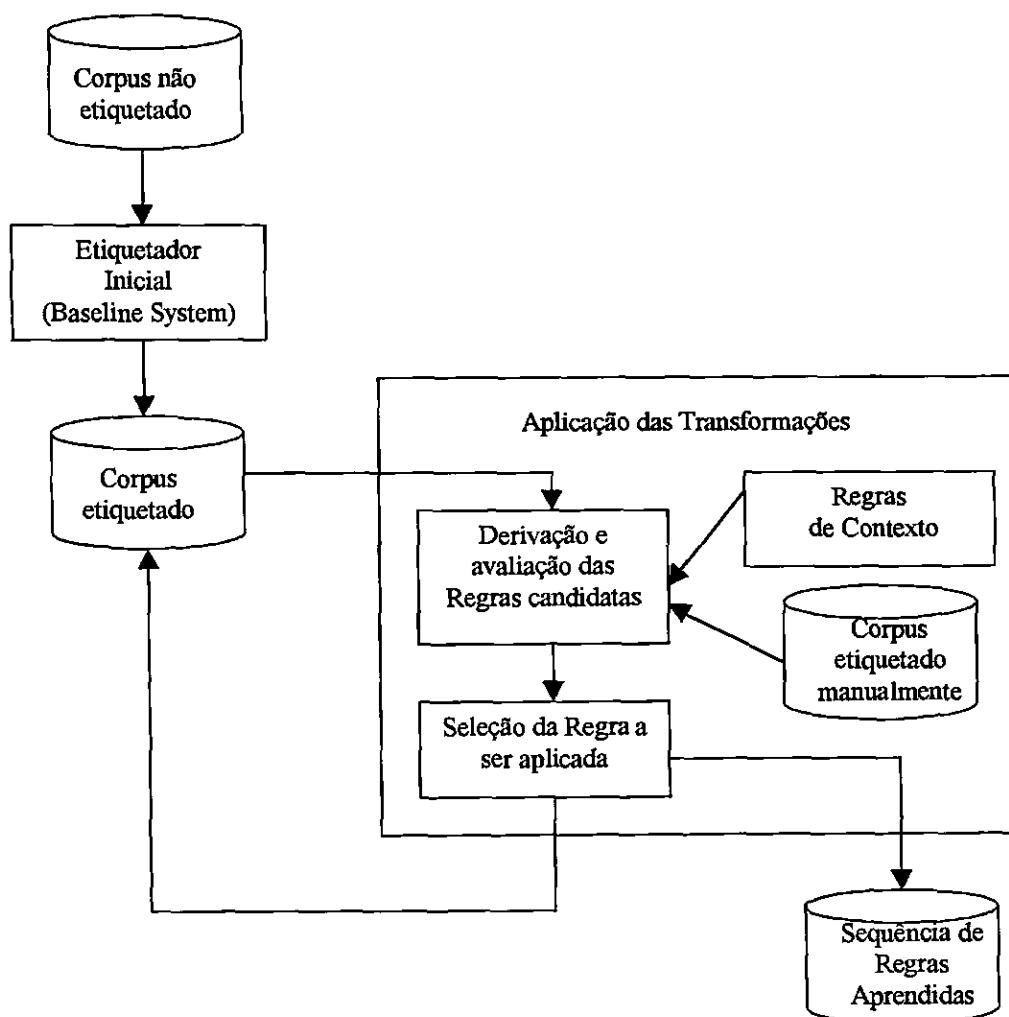


Figura 21 - Algoritmo de Aprendizado baseado em transformação dirigida por erro

3.3.1.1 O Algoritmo

Como mostrado na Figura 21, o corpus não-etiquetado passa primeiro por um processo de etiquetagem inicial. Esta é uma fase crucial para o sucesso do algoritmo e pode ser baseada em informações sobre freqüência de categorias para cada palavra de um léxico. Após ter sido etiquetado, o corpus é comparado com a etiquetagem correta (o corpus manualmente etiquetado) e uma lista de transformações é aprendida para ser aplicada à saída do etiquetador inicial. Cada transformação é composta pela regra de reescrita e pelo contexto que vai desencadear esta regra.

A cada interação de aprendizado, a transformação para melhorar a etiquetagem é encontrada de acordo com uma função objetivo utilizada e então adicionada à lista ordenada de transformações. O aprendizado continua até que não sejam mais encontradas transformações que possam melhorar o corpus etiquetado.

A Figura 22 mostra um exemplo de aprendizado de transformações. Neste exemplo assume-se que existem apenas quatro transformações possíveis (T_1 , T_2 , T_3 e T_4) e que a função objetivo utilizada seja o número total de erros. O corpus não etiquetado sofre a etiquetagem inicial, e o resultado é um corpus etiquetado com 5.100 erros. Em seguida, cada uma das transformações são aplicadas em ordem. Neste exemplo, nota-se que T_2 foi a transformação que possibilitou a maior redução de erros, sendo assim colocada como a primeira transformação da lista. É então aplicada a todo o corpus e o aprendizado continua. Nota-se que a transformação que mais diminuiu o número de erros foi T_3 , então T_3 é aprendida como a segunda transformação da lista. Após aplicar o etiquetador de estado inicial, T_2 e T_3 verifica-se que não há mais reduções no número de erros aplicando transformações, então o processo termina (seqüência mostrada na Figura 22).

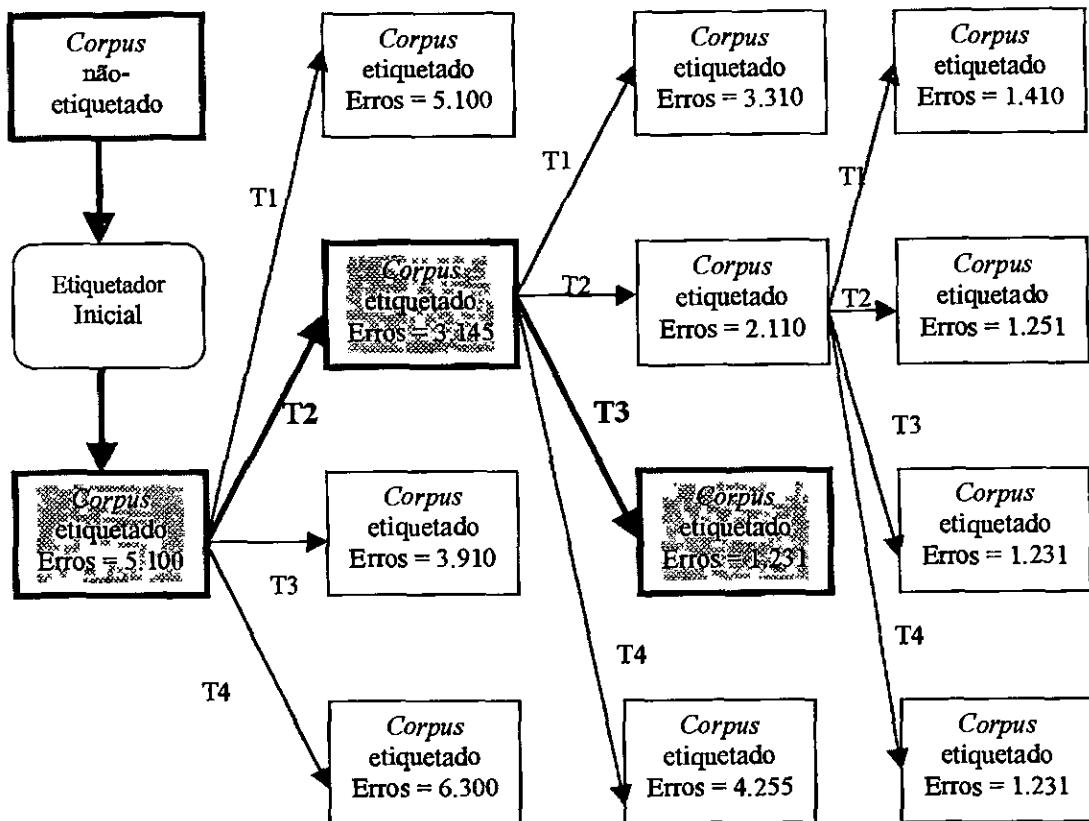


Figura 22 - Exemplo de Aprendizado Baseado em Transformação Dirigida por Erro (Brill,1994a)

3.3.1.2 O Etiquetador

O sistema é composta por dois módulos de etiquetagem: um inicial e um contextual. O inicial compila o corpus etiquetado, criando um léxico, em que para cada palavra há apenas sua etiqueta mais comum. Assim, na etiquetagem inicial, ele atribuirá a etiqueta mais provável para a palavra, conforme descrito no léxico (Figura 23).

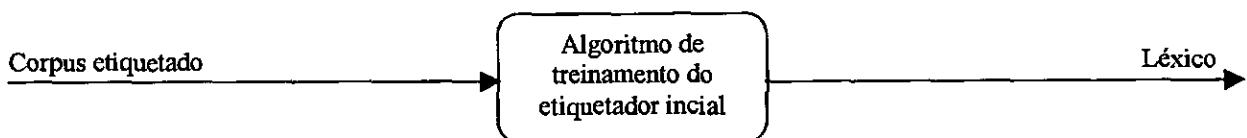


Figura 23 – Etiquetador baseado em Transformação dirigida por erro: Treinamento do etiquetador inicial

O Etiquetador inicial possui também um procedimento para a tratamento de palavras desconhecidas. Define que as palavras desconhecidas que iniciam por letra maiúscula tendem a ser substantivas próprios e o que as que se iniciam por letras minúsculas tendem a ser substantivo comum.

O Etiquetador contextual inferirá automaticamente as regras do contexto, a partir do corpus de treinamento etiquetado. Isto é feito fazendo-se a etiquetagem do corpus de treinamento usando a etiquetadora inicial e a seguir comparando-se automaticamente os resultados e gerando-se a lista de transformações. A partir da aplicação desta lista de transformações no corpus, serão geradas as regras de contexto. As aplicações que gerarem o melhor resultado serão utilizadas como regras de contexto (Figura 24).

Assim, obtém-se um conjunto de regras que analisa a atribuição das etiquetas feita pelo etiquetador inicial e as corrige conforme o contexto no qual as palavras aparecem.

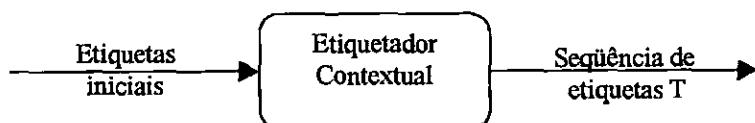


Figura 24 - Etiquetador baseado em transformação dirigida por erro: Etiquetador Contextual

As regras de contexto (transformações), alteram uma etiqueta de X para Y se:

- 1) A palavra não aparece no corpus de treinamento (Tabela 2)
- 2) A palavra foi etiquetada como Y pelo menos uma vez

As regras podem fazer referência a etiquetas anteriores/posteriores — regras não lexicalizadas (1 a 6) — ou a palavras anteriores/ posteriores — regras lexicalizadas (7 a 14) —, como mostrado na Tabela 1.

Tabela 1 - Regras de Contexto do Etiquetador Baseado em Transformação dirigida por erro

Mude de tag a para tag b quando:
1) A palavra anterior (posterior) foi etiquetada como z
2) A palavra duas posições antes (depois) foi etiquetada como z

3) Uma das duas palavras anteriores (posteriores) foi etiquetada como z.
4) Uma das três palavras anteriores (posteriores) foi etiquetada como z.
5) A palavra anterior foi etiquetada como z e a posterior como w.
6) A palavra anterior (posterior) foi etiquetada como z e a palavra "duas depois (antes)" foi etiquetada como w
7) A palavra que vem antes (depois) é w
8) A segunda palavra antes (depois) é w
9) Se uma das duas palavras que vem antes (depois) é w
10) Se a palavra atual é w e a anterior (posterior) é x
11) Se a palavra atual é w e a palavra anterior (posterior) é classificada z
12) Se a palavra atual é w
13) Se a palavra anterior (posterior) é w e a classificação anterior (posterior) é t
14) Se a palavra atual é w e a palavra anterior (posterior) é w2 e a classificação anterior (posterior) é t
<i>W e x se referem a todas as palavras no corpus de treinamento, e z se refere a todas as classes</i>

Tabela 2 - Regras para palavras desconhecidas

Mude a etiqueta de uma palavra desconhecida de X para Y se:
1. Apagando prefixo(sufixo) x, $ x \leq 4$, resulta em uma palavra (x é qualquer string de tamanho 1 a 4)
2. O primeiro (último) (1,2,3,4) caracteres da palavra é x
3. Adicionando a string x como prefixo (sufixo) resulta em uma palavra ($ x \leq 4$)
4. A palavra w nunca aparece imediatamente a esquerda(direita) da palavra
5. O caracter z aparece na palavra

Através de todo este processo o etiquetador consegue para a língua inglesa uma precisão de 97%, usando apenas 300 regras que foram automaticamente inferidas. Para tanto se utilizou um corpus de treinamento com 1 milhão de palavras (Brill, 1994a). Já o etiquetador simbólico para o inglês de Voutilainen (1995), descrito na Seção 3.1.1, possui 1.185 regras, além de mais 200 regras heurísticas opcionais.

4 ETIQUETADORES TREINADOS

Este capítulo mostra as decisões de projeto que foram tomadas neste trabalho com relação ao conjunto de etiquetas, corpus de treinamento e teste, abordagens de etiquetagem e métodos de avaliação (Seção 4.1). Traz também os experimentos realizados com cada etiquetador e seus resultados individuais (Seção 4.2).

4.1 Definições de projeto

Intellectuals solve problems; geniuses prevent them. - Albert Einstein

Ao se decidir construir um etiquetador deve-se ter em mente além da língua alvo, quais aplicações farão uso dos textos etiquetados. O requisito que norteou este projeto foi a construção de um etiquetador para o português contemporâneo do Brasil que possa ser utilizado para diversas aplicações, mas que em particular forneça uma boa base para o analisador sintático do ReGra. Baseando-se neste fato definimos o conjunto de etiquetas (Nilc tagset) que se encontra no apêndice A e os tipos de texto do corpus de treinamento e teste⁹.

Como os etiquetadores são utilizados em ambientes totalmente abertos e repletos de palavras novas e novas possibilidades de etiquetas para algumas palavras, houve uma preocupação

em encontramos nos artigos da literatura sobre técnicas para estimar a precisão real dos etiquetadores, isto é, como os etiquetadores se comportariam na etiquetagem de textos do mundo real.

4.1.1 Corpus de treinamento e teste

The things which hurt, instruct. - Benjamin Franklin

Os trabalhos descritos na literatura de etiquetagem da língua inglesa utilizam para treinamento e teste de um milhão a dois milhões de palavras, pois existem imensos corpora para esta língua. Existem muitos corpora etiquetados para o inglês, por exemplo: Brown Corpus, BNC, Wall Street Journal (parte do corpora Penn Treebank, contém 1.200.000 palavras). No entanto, para português não existe um corpus manualmente etiquetado desta magnitude e não existiam também corpus formados por diferentes tipos de texto (corpus mistos). O primeiro passo de projeto após a definição do conjunto de etiquetas foi a construção de um corpus de treinamento e teste misto que atendesse as necessidades do projeto ReGra, um corretor gramatical de uso geral e as demais aplicações que fazem uso de etiquetadores.

Nosso objetivo inicial era começar com um corpus manualmente etiquetado com cerca de 200.000 palavras cuja etiquetagem manual seria feita com o auxílio de uma ferramenta semi-automática de auxílio à etiquetagem e em cerca de seis meses conseguir um corpus de 1 milhão de palavras. Dado que a correção da etiquetagem automática mostra ser mais rápida que a etiquetagem manual (Marcus et al., 1993), o processo a ser seguido seria chegar aos 1 milhão de palavras através de um processo incremental e cíclico de treinamento — etiquetagem e correção de 200 em 200 mil palavras. Devido à dificuldade do processo, não conseguimos atingir esta marca até o momento de escrita desta dissertação.

Para compor nosso corpus de treinamento e teste selecionamos textos do corpora do Nilc pertencentes a três gêneros: didático, jornalístico e literário. Um dos objetivos deste trabalho é avaliar os etiquetadores por gêneros. A escolha destes três gêneros foi feita para se abranger em particular:

⁹ O corpus de treinamento e teste se encontra disponível em (Disponível em <http://www.nilc.icmc.sc.usp.br/tools.html/TAGCORPUS>

- 1) textos simples, isto é, aqueles que seguem uma estrutura formal fixa, por exemplo a escrita técnica (didáticos);
- 2) textos mais próximos da linguagem viva (jornalístico);
- 3) textos com estrutura livre, isto é, com formas menos comuns como ordem inversa por exemplo, (literários).

Barthes (1986) deixa mais clara a diferença entre estes tipos de textos definindo a escritura de grau zero, a escritura neutra e a escritura literária. Segundo Barthes, a escritura de grau zero seria uma escritura indicativa, amodal, em suas palavras uma escritura de jornalista, mas que às vezes faz uso de estruturas optativas. A escritura neutra seria uma espécie de língua básica, distanciada por igual das linguagens vivas e da linguagem literária propriamente dita, ou seja, um texto simples sem coloquialismos, regionalismos e neologismos. A escritura literária seria composta pelos textos literários que conhecemos, uma estrutura livre que abrange desde a prosa clássica a poesia moderna.

Além dos genêros de texto que farão parte do corpus, temos também que decidir se serão ou não mantidos títulos, frases entre parênteses e resumos nos textos. No nosso caso não mantivemos os títulos, mas mantivemos os textos entre parênteses e não precisamos nos preocupar com resumos, que não apareciam nos textos escolhidos. Já Kim & Norgard (1998), na preparação do corpus para ser etiquetado para o uso no processo de construção de dicionários de associações baseadas em sintagmas nominais para mapeamento de vocabulário, mantém os títulos e excluem textos entre parênteses dizendo que é muito difícil de tratar todas as variações de textos que podem aparecer entre parênteses, independente do algoritmo utilizado.

Tinhamos, assim, um corpus de treinamento e teste com cerca de 200.000 palavras com um terço de textos didáticos, um terço de textos jornalísticos e um terço de textos literários. No entanto, no final do prazo em que imaginamos que teríamos 1 milhão de palavras, dezembro de 1999, tínhamos apenas cerca de 100.000 palavras etiquetadas. Fato que ocorreu em parte por termos subestimado a tarefa de etiquetagem, em parte devido às várias alterações do NILC tagset (quatro) que acarretaram em remarcações do corpus e em parte pela falta de uma pessoa qualificada para cuidar unicamente da tarefa de verificação de etiquetagem.

Com todos estes contratemplos, em 20 de julho de 2000, obtivemos a última versão do nosso corpus de treinamento e teste contendo 104.963 palavras, que foi utilizada em todos os

experimentos descritos neste trabalho. Como o corpus manual é menor do que o planejado de 200.000 palavras não mantivemos a proporção desejada de cada um dos tipos de texto no corpus como mostra a Tabela 3.

Tabela 3 - Corpus de treinamento e teste

Tipo de Corpus	Tamanho do corpus
D – Didático	16.256 palavras
J – Jornalístico	56.653 palavras
L – Literário	32.054 palavras

4.1.1.1 Etiquetagem manual de um corpus de treinamento e teste

Quem se mete em atalhos não se livra de trabalhos – Ditado popular

Uma atividade importante quando da construção de um corpus manualmente etiquetado é o acompanhamento durante todo o processo de etiquetagem para que se possa ter estimativas de quão consistente será o corpus no final do processo, ou seja, qual será a taxa de erro da etiquetagem manual. Outra alternativa é fazer experimentos iniciais e generalizar sobre os valores obtidos.

No projeto do Penn Treebank (Marcus et al., 1993) por exemplo, foram feitos experimentos para se estimar qual seria a consistência do corpus manualmente etiquetado por linguistas e corrigido por linguistas após a etiquetagem automática. Nestes experimentos quatro linguistas estavam envolvidos com a etiquetagem de oito textos com 2.000 palavras cada selecionados do Brown Corpus, sendo que quatro seriam manualmente etiquetados e quatro seriam apenas corrigidos. Fizeram então experimentos para avaliar a taxa de discordância entre cada par de linguistas. A taxa ficou em torno de 7.2% nos textos manualmente etiquetados e 4.1% nos textos corrigidos. E a média de discordância entre o que foi feito por cada linguista e os benchmarks (cuidadosamente corrigidos) ficou em torno de 5.7% na tarefa de etiquetagem e 3.4% na tarefa de correção.

Neste projeto, não foram feitos os experimentos iniciais para se estimar a consistência dos dois tipos de etiquetagem (etiquetados manualmente e com correção manual), mas estimamos a

taxa de erro da etiquetagem manual de outra forma. Foi feita uma seleção de trechos dentro de cada tipo para que fossem assim corrigidos cuidadosamente palavra por palavra e pudessemos assim estimar a taxa em todo o corpus. A Tabela 4 mostra as taxa de erro em cada trecho e a estimativa para todo o corpus.

Tabela 4 - Taxa de erro na etiquetagem manual

	Tamanho	Número de erros no trecho	Número de erros estimado no corpus	Taxa de erro estimada no corpus
Trecho formado por períodos de textos didáticos	322 palavras	5 ¹⁰	252	1,55%
Trecho formado por períodos de textos jornalísticos	927 palavras	30 ¹¹	1833%	3,23%
Trecho formado por períodos de textos literários	486 palavras	14 ¹²	923%	2,88%
Todo o corpus	3008		2,86%	

¹⁰ Das 5 palavras com etiquetagem errada duas estão como CONCOORD (existem 8 destas no trecho e 456 em todos os textos didáticos), uma está como PD (existem 6 destas no trecho e 130 em todos os textos didáticos), uma está como PPS (existem 6 destas no trecho e 56 em todos os textos didáticos) e uma está como VTD (existem 24 destas no trecho e 705 em todos os textos didáticos).

¹¹ Das 30 palavras com etiquetagem errada uma está como VINT (existem 6 destas no trecho e 562 em todos os textos jornalísticos), duas estão como LCONJ (existem 8 destas no trecho e 247 em todos os textos jornalísticos), duas estão como ART (existem 77 destas no trecho e 4398 em todos os textos jornalísticos), três estão como CONJSUB (existem 8 destas no trecho e 580 em todos os textos jornalísticos), três estão como ADV (existem 31 no trecho, 1869 em todos os textos jornalísticos), uma está como VAUX (existem 8 no trecho e 779 em todos os textos jornalísticos), cinco estão como VTD (existem 60 no trecho e 3523 em todos os textos jornalísticos), duas estão como VTI (existem 6 no trecho e 779 em todos os textos jornalísticos), uma está como PREP+ART (existem 60 no trecho e 3715 em todos os textos jornalísticos), três estão como CONCOORD (existem 40 no trecho e 1532 em todos os textos jornalísticos), uma está como PR (existem 102 no trecho e 598 em todos os textos jornalísticos), duas estão como N (existem 214 no trecho e 12100 em todos os textos jornalísticos), uma está como ADJ (existem 69 no trecho e 3596 em todos os textos jornalísticos), duas estão como PREP (existem 91 no trecho e 5082 em todos os textos jornalísticos) e uma está como PAPASS (existem 2 no trecho e 29 em todos os textos jornalísticos).

¹² Das 14 palavras com etiquetagem errada uma está como PREP+ART (existem 27 no trecho e 1784 em todos os textos literários em todos os textos literários), duas estão como VINT (existem 6 no trecho e 767 em todos os textos literários), três estão como PREP (existem 33 no trecho e 2293 em todos os textos literários), uma está como ADJ (existem 40 no trecho e 2406 em todos os textos literários), uma está como VAUX (existem 5 no trecho e 261 em todos os textos literários), uma está como ADV (existem 28 no trecho e 1426 em todos os textos literários), duas estão como VTD (existem 27 no trecho e 1727 em todos os textos literários), uma está como N (existem 100 no trecho e 6214 em todos os textos literários) e uma está como VBI (existem 3 no trecho e 173 em todos os textos literários).

4.1.2 Abordagens de etiquetagem

O ideal deve, como a árvore, ter suas raízes na terra - Graf

Para o português contemporâneo do Brasil o primeiro passo foi fazer um estudo comparativo que reunisse pelo menos um bom etiquetador por abordagem: um estatístico, um neural e um híbrido. Escolhemos os etiquetadores dentre os que são independentes da língua e que estavam disponíveis na WWW. Tentando obter dentre as possibilidades em cada abordagem o que tivesse maior precisão geral para o inglês e/ou tivesse sido utilizado em vários experimentos para outras línguas.

De acordo com este critério, escolhemos para representar a abordagem estatística os etiquetadores TreeTagger (Schmid, 1995) e MXPOST (Ratnaparkhi, 1996), para representar a abordagem neural o etiquetador elástico (Ma et al., 1999) e para representar a abordagem híbrida o etiquetador TBL (Brill 1994a). Outra meta de pesquisa foi elaborar um etiquetador simbólico descrito em detalhes na Seção 4.2.1.4.

4.1.3 Avaliação

O maior objetivo de um classificador é ser capaz de predizer com sucesso a respeito de novos casos.

Na literatura de etiquetagem morfossintática quando se questiona a qualidade de um etiquetador os critérios levantados são:

- sua precisão geral (*accuracy*), que é dada pelo número de palavras classificadas corretamente dividido pelo número de palavras do seu arquivo de teste;
- se o etiquetador é independente da língua, ou seja, se pode ou não ser treinado para outras línguas;
- seu tempo de etiquetagem e tempo de treinamento;
- qual o formato de entrada exigido;
- se existem restrições quanto ao tamanho do conjunto de etiquetas, tanto para a precisão quanto para a complexidade do algoritmo de treinamento;
- se existe a possibilidade de incrementar o léxico, isto é, fazer um treinamento incremental.

Dentre estas seis questões, apenas as questões 1 e 3 exigem experimentos. A questão 1 é a de resposta mais difícil já que é desejado que o etiquetador tenha bons resultados em dados

reais – e não apenas no *corpus* de teste. Apesar disso, em nossa revisão da literatura de etiquetagem morfossintática não encontramos artigos que mostrassem técnicas para estimar qual seria o desempenho do etiquetador frente a dados reais. Na seção 4.1.3.1 discutiremos algumas técnicas baseadas na teoria estatística de *resampling* que foram utilizadas para estimar a taxa de erro verdadeira de classificadores. Em geral, no entanto, nosso objetivo não é apenas avaliar a qualidade de um etiquetadores, mas também avaliar as influências de diferentes fatores em seu desempenho – quão diferentes são os resultados de etiquetadores que utilizam diferentes abordagens de etiquetagem (Seção 4.2.1), qual a influência do conjunto de etiquetas (Seção 4.2.2), e como diferentes tipos de texto podem influenciar um etiquetador (Seção 4.2.3) – para este tipo de avaliação nos baseamos nas recomendações do EAGLES (1996). Este tipo de avaliação está descrito na Seção 4.1.3.2.

4.1.3.1 Avaliação da taxa de erro verdadeira

A taxa de erro, que é a forma mais utilizada para medir o desempenho de um classificador, pode ser calculada através da equação:

$$\text{Taxa de erro} = \frac{\text{número de erros}}{\text{número de casos}}$$

E pode ser de dois tipos:

- 1) Taxa de erro aparente: é calculada utilizando-se somente os exemplos de treinamento, ou seja, inicialmente o sistema é treinado com um conjunto de exemplos e depois é verificado quantos erros o classificador cometeu classificando exemplos que fazem parte dos mesmos do treinamento.
- 2) Taxa de erro verdadeira: é a taxa de erro obtida sobre um número muito grande de novos casos, selecionados independentemente dos casos usados para treinar o classificador.

A *taxa de erro verdadeira* é uma excelente medida para a taxa de erro de um classificador, no entanto, para muitas aplicações não se tem um número grande de exemplos, o que inviabiliza

seu cálculo. Mas através de técnicas estatísticas que apresentam de diferentes formas os exemplos ao classificador é possível obter aproximações da taxa de erro verdadeira bem mais confiáveis que a taxa de erro aparente (Dietterich, 1997).

Um dos requisitos para se estimar a taxa de erro verdadeira é manter a amostra de exemplos em ordem aleatória, ou seja, a amostra de exemplos não deve ser pre-selecionada, de forma a evitar que seja feita qualquer suposição sobre a qualidade dos exemplos. Partindo deste princípio, temos duas técnicas para estimar a taxa de erro verdadeira: *Probably Approximately Correct* (PAC), estimativa da taxa de erro através do paradigma treinar-e-testar.

Nos casos em que estão disponíveis um número ilimitado de casos para testar e treinar, a taxa de erro aparente é a taxa de erro verdadeira. Mas, como em geral existe um número de casos limitado, questionou-se quantos casos seriam necessários para que a taxa de erro aparente se tornasse efetivamente a taxa de erro verdadeira dadas uma amostra e uma taxa de erro relativamente baixa. Resultados teóricos mostraram que, tipicamente, a taxa de erro em novos casos não supera em duas vezes a taxa de erro para os exemplos da amostra. A análise *Probably Approximately Correct* (Kearns, 1994) surge dessa idéia. É uma análise de pior caso, e os resultados mostram que é necessário um grande número de casos para garantir o desempenho, já que é uma técnica para estimar a taxa de erro geral e não para uma população em particular.

No nosso caso, por não termos um conjunto grande de exemplos temos de recorrer a técnicas que dada uma amostra estimem a taxa de erro para uma população e não para todas as possíveis populações, o que requer muito menos casos já que uma única distribuição da população é considerada. Uma opção é o paradigma treinar-e-testar, em que os casos são particionados em dois grupos, um para treinar e outro para testar. Esta técnica de análise não dá garantias para todas as possíveis distribuições, mas fornece uma estimativa da taxa de erro verdadeira para a população considerada. O princípio básico do paradigma treinar-e-testar é dividir a amostra de exemplos em dois grupos mutuamente exclusivos — conjunto de treinamento (utilizado exclusivamente para treinar) e conjunto de teste (utilizado exclusivamente para testar). Os dois conjuntos de casos devem ser amostras aleatórias de alguma população e devem ser independentes. Dizer que devem ser independentes significa que a única relação entre eles é o fato de pertencerem a uma mesma população. Para isso devem ser coletados em datas diferentes, ou por pesquisadores diferentes. Resultados práticos da literatura mostram que esta independência gera excelentes aproximações da taxa de erro mesmo quando as amostras de

exemplos são pequenas. Esta técnica tem cerca de 95% de confiabilidade, ou seja, não há mais de 5% de probabilidade de que a taxa de erro exceda os valores apresentados – estes valores são de considerações básicas de probabilidade e estatística (Batista & Monard, 1998).

Existem várias formas de dividir a amostra em conjunto de treinamento e teste. Uma delas é o *método Holdout* ou *método H* que divide a amostra em 2/3 para treinamento e 1/3 para teste e quando o número de casos é maior que 1.000 casos costuma aumentar a fração de treinamento. Este método apresenta uma boa estimativa dadas amostras grandes, entretanto para amostras pequenas apresenta aproximações pessimistas. Para os casos em que o conjunto de exemplos não é grande são utilizadas variações de treinar-e-testar conhecidas como métodos de resampling.

Os métodos de resampling consistem em realizar vários experimentos de treinar-e-testar com diferentes partições de exemplos. Alguns exemplos destes métodos são: *Random Subsampling*, *Cross-validation* e *Bootstrapping*.

O método *Random Subsampling* faz vários experimentos treinar-e-testar gerando várias partições de treinamento e teste, solucionando assim o problema de selecionar um conjunto de teste que não seja representativo o que podia acontecer no Holdout. Isto é feito gerando um número de partições menor ou igual ao número de casos, em que o conjunto de teste é composto do que não foi utilizado na formação do conjunto de treinamento, como mostra a Tabela 5.

Tabela 5 - Comparação entre Holdout e Random Subsampling (Batista & Monard, 1998)

	<i>Holdout</i>	<i>Random subsampling</i>
Casos de Treinamento	j	j
Casos de Teste	$n - j$	$n - j$
Iterações	1	Número de interações $\ll n$

O método *Cross-validation* é também conhecido como *k-fold Cross-validation*, sendo k o número de partições geradas aleatoriamente a partir da amostra de exemplos para treinar-e-testar o sistema, sendo que a amostra de exemplos é dividida em k partições mutuamente exclusivas. A cada iteração uma partição diferente é utilizada para testar o sistema e todas as outras $k-1$

iterações são utilizadas para treinar o sistema. A taxa de erro é a média das taxas de erro calculadas dadas as diversas partições.

Um tipo especial de *Cross-validation* é o *Leaving-one-out* em que para uma amostra de n exemplos são feitas n iterações, sendo que em cada iteração um exemplo é retirado para testar e os demais $n - 1$ são utilizados para treinar. A taxa de erro é calculada dividindo-se o número de erros observados por n . A estimativa da taxa de erro verdadeira do *Leaving-one-out* é praticamente não tendenciosa e com vários conjuntos de exemplos tende a taxa de erro verdadeira, porém é um algoritmo muito caro. Por ser um algoritmo caro costuma ser utilizado apenas para amostras realmente pequenas e em casos de amostras maiores geralmente é utilizado o *K-fold Cross-validation* com k igual a 10.

Apesar de quase não ser tendencioso, o *Leaving-one-out* possui alta variância, portanto, para amostras pequenas em que a variância tende a dominar, métodos com baixa variância tendem a apresentar melhores resultados. Um destes métodos que tem sido bastante estudado na área de estatística aplicada é o *Bootstraping*, sendo que dentre os estimadores *Bootstraping* existentes dois se destacam: o *e0 Bootstraping* e o *.632 Bootstraping*. No caso do estimador *e0*, o conjunto de treinamento consiste de n casos copiados aleatoriamente da amostra inicial – ou seja, continuam a fazer parte da amostra inicial de forma que o conjunto de treinamento poderá conter casos repetidos. E o conjunto de teste é formado por todos os casos que não fizerem parte do conjunto de treinamento. Em geral são feitas cerca de 200 interações. A aproximação da taxa de erro verdadeira é a média das taxas calculadas. Esta técnica possui uma fração média de casos não repetidos no conjunto de treinamento que é de .0632, e uma fração média de casos não repetidos no conjunto de teste de 0.368. O estimador *.632 Bootstraping* faz uso desta informação e sua estimativa da taxa de erro verdadeira é:

$$\text{.368 * taxa de erro aparente para todos os casos de treinamento e teste} + \text{.632 * e0}.$$

Neste trabalho foi utilizado apenas o estimador *e0 Bootstraping* para estimar a taxa de erro verdadeira do etiquetador TreeTagger. Quando vários experimentos de treinar-e-testar são realizados, um novo classificador é projetado com cada conjunto de treinamento, esta foi a razão de termos escolhido o TreeTagger para este experimento, dado que é o etiquetador de treinamento mais rápido. Todos os algoritmos implementados neste trabalho para auxiliar no processo de etiquetagem, na avaliação de etiquetados, na combinação de etiquetadores

(Capítulo 5) e na avaliação da combinação fazem parte de um software, chamado InCorpora descrito no Apêndice D.

4.1.3.2 Fatores de impacto no processo de etiquetagem

Os fatores de impacto no processo de etiquetagem são três: a abordagem de etiquetagem, o conjunto de etiquetas e os tipos de textos utilizados no treinamento e teste (EAGLES, 1996). Desta forma é essencial para um bom estudo avaliar o impacto de:

- 1) Diferentes métodos de etiquetagem nos resultados, comparando os resultados de diferentes etiquetadores – as diferenças na precisão e os erros cometidos por cada método. Esta avaliação serve para indicar também quais são as etiquetas mais problemáticas para cada etiquetador, quais são as etiquetas que são confundidas da mesma forma em todos os etiquetadores testados e quais são os contextos problemáticos para todos os etiquetadores.
- 2) Diferentes conjuntos de etiquetas nos resultados da etiquetagem, objetivando encontrar uma forma de modificar o conjunto de etiquetas para atingir melhores resultados, e através da análise dos resultados encontrar dicas para possíveis casos difíceis de tratar na construção do conjunto de etiquetas.
- 3) Diferentes tipos de textos com o objetivo de encontrar formas de medir o impacto de diferenças que possam ocorrer entre textos de treinamento e teste, já que alguns tipos de texto (por exemplo: jornalísticos, jurídicos e técnicos) diferem entre si na distribuição das construções sintáticas e isso pode conduzir a ligeiras diferenças entre os modelos estatísticos de cada tipo.

Neste trabalho a avaliação 2 foi feita apenas para verificar o peso de termos em nosso conjunto de etiquetas aquelas que consideram a transitividade, e não com o objetivo de encontrar um conjunto de etiquetas ideal.

4.2 Resultados dos experimentos com os etiquetadores individuais

Personally, I'm always ready to learn, although I do not always like being taught. - Winston Churchill

O *corpus* de treinamento e teste manualmente etiquetado foi dividido de diferentes formas pois realizamos diferentes experimentos que são apresentados nesta seção e no Capítulo 5. A divisão padrão do corpus foi de 80% para treinamento (C3, C4, C5 e C6), 10% para o aprendizado dos algoritmos de combinação (C31, C41, C51, C61 e C7) – chamado de corpus de calibração e 10% para testes de precisão de cada etiquetador e dos algoritmos de combinação (C32, C42, C52, C62 e C8) – chamado corpus de teste. O corpus de treinamento é formado por 80% dos textos de cada gênero (C3) ou 80% dos textos de um gênero quando da avaliação dos tipos de texto (C4, C4 e C6). Para verificar o efeito que o uso de um corpus maior no treinamento tem na precisão foi feita também uma divisão do corpus manualmente etiquetado em 90% para treinamento (C2) e 10% para teste (C21). Para avaliar o efeito das palavras desconhecidas na precisão foi feita também a divisão em 100% para treinamento (C1) e 10% para teste (C21). Foram construídos também seis conjuntos de calibração e seis de teste através dos conjuntos C41, C51 e C61 e, C42, C52 e C62 respectivamente, tomados dois a dois. Cada um destes conjuntos tem seu tamanho proporcional ao conjunto de textos de treinamento do tipo de texto que foi deixado de fora em sua formação, e tais conjuntos servirão para verificarmos os efeitos de termos tipos de texto diferentes formando o corpus de treinamento e teste. A Tabela 6 mostra a divisão do corpus em Kb, número de períodos e palavras.

Tabela 6 - Divisões do corpus

Corpus	Tamanho
C1 - Corpus total	960 Kb – 4957 períodos – 104.963 palavras
C2 - Corpus de treinamento (90%)	863 Kb – 4478 períodos – 94.472 palavras
C21 - Corpus de teste (90%)	98 Kb – 479 períodos – 10.491 palavras
C3 - Corpus de treinamento (80%)	768 Kb – 4001 períodos – 83.972 palavras
C31 -Corpus de calibração (80%)	95 Kb – 477 períodos – 10.500 palavras
C32 - Corpus de teste (80%)	98 Kb – 480 períodos – 10.491 palavras
C4 – Conjunto de textos didáticos de treinamento	124 Kb – 589 períodos – 13.004 palavras
C41 - Conjunto de textos didáticos de calibração	16 Kb – 68 períodos – 1.635 palavras
C42 - Conjunto de textos didáticos de teste	16 Kb – 64 períodos – 1.617 palavras
C5 – Conjunto de textos jornalísticos de treinamento	413 Kb – 2040 períodos – 45.327 palavras
C51 - Conjunto de textos jornalísticos de calibração	52 Kb – 229 períodos – 5.661 palavras
C52 - Conjunto de textos jornalísticos de teste	53 Kb – 241 períodos – 5.665 palavras
C6 – Conjunto de textos literários de treinamento	232 Kb – 1322 períodos – 25.641 palavras
C61 - Conjunto de textos literários de calibração	29 Kb – 180 períodos – 3.204 palavras
C62 - Conjunto de textos literários de teste	31 Kb – 175 períodos – 3.209 palavras
C7 - Conjuntos de textos de 2 gêneros de calibração JL	15 Kb – 71 períodos – 1.611 palavras

C8 - Conjuntos de textos de 2 gêneros de teste JL	15 Kb – 75 períodos – 1.606 palavras
C9 - Conjuntos de textos de 2 gêneros de calibração DL	44 Kb – 248 períodos – 4.839 palavras
C10 - Conjuntos de textos de 2 gêneros de teste DL	46 Kb – 239 períodos – 4.826 palavras
C11 - Conjuntos de textos de 2 gêneros de calibração DJ	30 Kb – 116 períodos – 3.222 palavras
C12 - Conjuntos de textos de 2 gêneros de teste DJ	30 Kb – 130 períodos – 3.193 palavras

Nosso corpus possui uma taxa de ambigüidade em torno de 54,32%, sendo que o conjunto de textos didáticos contém 14,77% das palavras ambíguas, o conjunto de textos jornalísticos 53,69% e o conjunto de textos literários 31,54%, como mostrado na Figura 25. A Tabela 7 mostra a taxa de ambigüidade em cada uma das divisões do corpus. Para verificar se esta taxa alta de ambigüidade não acontecia apenas nos textos utilizados neste trabalho, gerou-se uma lista de palavras ambíguas a partir da Base de dados lexicais do Nilc (BDL) (1.515.500 palavras), o que resultou em uma lista com 19.817 palavras ambíguas. Esta lista serviu para realizar uma busca em parte do corpora do NILC (8.844.492 de palavras), na qual apareceram 13.950 das palavras da lista, que resultam em uma taxa de ambigüidade neste corpora de 44,49%

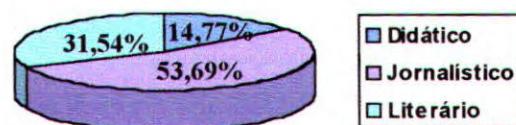


Figura 25 – Porcentagem da taxa de ambigüidade do corpus por conjunto de textos

Tabela 7 - Taxas de ambigüidade nas divisões do corpus

Corpus	Taxa de ambigüidade
C1 - Corpus total	54,32%
C2 - Corpus de treinamento	54,33%
C21 - Corpus de teste	54,30%
C3 - Corpus de treinamento	54,44%
C31 -Corpus de calibração	53,42%
C32 - Corpus de teste	54,30%
C4 – Conjunto de textos didáticos de treinamento	51,47%
C41 - Conjunto de textos didáticos de calibração	51,38%
C42 - Conjunto de textos didáticos de teste	54,98%
C5 – Conjunto de textos jornalísticos de treinamento	53,79%
C51 - Conjunto de textos jornalísticos de calibração	55,04%
C52 - Conjunto de textos jornalísticos de teste	55,02%
C6 – Conjunto de textos literários de treinamento	57,1%
C61 - Conjunto de textos literários de calibração	51,59%
C62 - Conjunto de textos literários de teste	52,69%
C7 - Conjuntos de textos de 2 gêneros de calibração JL	52,45%
C8 - Conjuntos de textos de 2 gêneros de teste JL	56,23%
C9 - Conjuntos de textos de 2 gêneros de calibração DL	51,52%
C10 - Conjuntos de textos de 2 gêneros de teste DL	53,46%
C11 - Conjuntos de textos de 2 gêneros de calibração DJ	54,06%
C12 - Conjuntos de textos de 2 gêneros de teste DJ	56,22%

Outra estatística importante é saber qual a porcentagem de palavras desconhecidas nos corpus de calibração e corpus de teste com relação ao corpus de treinamento, ou seja, as palavras que apareceram nestes corpora que não faziam parte do treinamento (1) ou que faziam mas com outras etiquetas (2). Estes dados são mostrados na Tabela 8.

Tabela 8 - Porcentagem de palavras desconhecidas

Corpus	Porcentagem de palavras desconhecidas (1)	Porcentagem de palavras desconhecidas (2)
C21 - Corpus de teste	14,20% com relação a C2	0,18% com relação a C2
C31 -Corpus de calibração	15,23% com relação a C3	0,26% com relação a C3
C32 - Corpus de teste	15,21% com relação a C3	0,20% com relação a C3
C41 - Conjunto de textos didáticos de calibração	13,45% com relação a C4	0,49% com relação a C4
C42 - Conjunto de textos didáticos de teste	14,04% com relação a C4	0,37% com relação a C4
C51 - Conjunto de textos jornalísticos de calibração	15,95% com relação a C5	0,32% com relação a C5
C52 - Conjunto de textos jornalísticos de teste	14,88% com relação a C5	0,26% com relação a C5
C61 - Conjunto de textos literários de calibração	26,99% com relação a C6	0,25% com relação a C6
C62 - Conjunto de textos literários de teste	28,51% com relação a C6	0,25% com relação a C6
C7 - Conjuntos de textos de 2 gêneros de calibração JL	35,19% com relação a C4	0,37% com relação a C4
C8 - Conjuntos de textos de 2 gêneros de teste JL	31,75% com relação a C4	0,37% com relação a C4
C9 - Conjuntos de textos de 2 gêneros de calibração DL	22,96% com relação a C5	0,33% com relação a C5
C10 - Conjuntos de textos de 2 gêneros de teste DL	24,22% com relação a C5	0,29% com relação a C5
C11 - Conjuntos de textos de 2 gêneros de calibração DJ	27,13% com relação a C6	0,43% com relação a C6
C12 - Conjuntos de textos de 2 gêneros de teste DJ	24,15% com relação a C6	0,41% com relação a C6

4.2.1 Avaliação dos métodos de etiquetagem

Pelos frutos se conhece a árvore. - Provérbio

A avaliação dos métodos de etiquetagem pode ser dividida em metas mais precisas. Escolhemos oito metas mostradas abaixo, cujos resultados aparecem nas próximas subseções:

- Verificar o tempo gasto por cada etiquetador com treinamento e etiquetagem
- Calcular a precisão geral de cada etiquetador
- Utilizar o algoritmo *e0 Bootstrapping* para estimar a taxa de erro do etiquetador de treinamento mais rápido
- Verificar quais eram as etiquetas mais problemáticas para cada etiquetador
- Averiguar se existia um etiquetador que apresentasse uma melhor precisão dado o nosso corpus e conjunto de etiquetas
- Averiguar se existiam etiquetas que eram problemáticas para todos os etiquetadores

- Checar se haveria um acréscimo na precisão do etiquetador de maior precisão com o aumento do corpus de treinamento através do treinando deste com 90% do corpus.
- Verificar a importância das palavras desconhecidas verificando se a precisão geral do etiquetador de maior precisão sofreria acréscimo quando com treinamento fechado, ou seja, quando 100% do corpus fosse utilizado para treinamento de forma que o teste não conteria palavras desconhecidas

4.2.1.1 TreeTagger

O TreeTagger permite a parametrização do contexto, tendo sido bastante utilizado na literatura como trígrama. Assim, nossos primeiros experimentos com o TreeTagger foram feitos usando um contexto de tamanho três. Como o manual do sistema sugeria que para conjuntos de etiquetas grandes ou corpus de treinamento pequeno a precisão geral poderia melhorar com a alteração do contexto para um bigrama este foi nosso segundo experimento. Entretanto, como mostra a Tabela 9 isto não aconteceu. O melhor resultado foi o do etiquetador como trígrama: 88,47%. Utilizamos o TreeTagger como unígrama para servir de base para comparações. O TreeTagger, tanto como bigrama quanto como trígrama, foi o etiquetador que apresentou menor tempo de treinamento e etiquetagem rodando em uma Sun Ultra 1 com 128 Mb de RAM. Os tempos gastos por ele no treinamento e etiquetagem são mostrados na Tabela 10. O treinamento do TreeTagger como trígrama resultou em um modelo com 2163 nós e uma altura máxima da árvore igual a 73.

Utilizando o algoritmo de *eD Bootstrapping* com 200 iterações, a taxa de erro verdadeira estimada para o etiquetador TreeTagger como trígrama foi de 88,95%. O processo de estimar a taxa de erro verdadeira para o TreeTagger ~ 200 treinamentos e etiquetagens – durou 3 horas e 10 minutos.

Tabela 9 - Precisão Geral do etiquetador TreeTagger

	Resultados da precisão
TreeTagger-Unígrama (C3)	
Corpus de calibração (C31)	79,62%
Corpus de teste (C32)	80,01%
TreeTagger-Bigrama (C3)	
Corpus de calibração (C31)	87,77%
Corpus de teste (C32)	88,41%
TreeTagger-Trígrama (C3)	

Corpus de calibração (C31)	88.01%
Corpus de teste (C32)	88.47%

Tabela 10 - Tempos de treinamento e etiquetagem - TreeTagger

	Tempo de treinamento	Tempo de etiquetagem
TreeTagger-Unigrama (C3)	-	-
Corpus de calibração (C31)	-	menos de 1 segundo
Corpus de teste (C32)	-	menos de 1 segundo
TreeTagger-Bigrama (C3)	6 segundos	-
Corpus de calibração (C31)	-	menos de 1 segundo
Corpus de teste (C32)	-	menos de 1 segundo
TreeTagger-Trígrama (C3)	51 segundos	-
Corpus de calibração (C31)	-	menos de 1 segundo
Corpus de teste (C32)	-	menos de 1 segundo

Analizando a precisão por etiquetas nota-se que as etiquetas problemáticas¹³ para o TreeTagger correspondem a 33,33% das etiquetas como mostra a Tabela 11, em que as etiquetas problemáticas aparecem em vermelho. A Figura 26 mostra de forma mais clara as precisões, agrupando algumas delas¹⁴.

Tabela 11 - Precisão por etiqueta no corpus de teste – TreeTagger como trígrama

Etiquetas	Precisão	Etiquetas	Precisão
ADJ	82,60%	PREP+PPOA	- ¹⁵
ADV	81,07%	PREP+ADV	100%
ART	97,75%	PPOA+PPOA	-
NC	93,23%	ADV+PPR	-
ORD	81,25%	ADV+PPOA	0%
NO	0%	ADJ+PPOA	-
N	93,39%	VTD+PPOA	6,49%
NP	87,91%	VTD+PAPASS	0%
CONCOORD	94,50%	VAUX+PPOA	100%
CONJSUB	71,70%	VBI+PPOA	0%
PD	92,31%	VLIG+PPOA	0%

¹³ Consideramos como problemáticas as etiquetas para as quais a precisão é menor que 80%.

¹⁴ Num – são as etiquetas referentes aos numerais - NC, ORD e NO

Sub – são as etiquetas referentes a substantivo – N e NP

Conj – são as etiquetas referentes as conjunções – CONCOORD e CONJSUB

Pron – são as etiquetas referentes aos pronomes – PD, PIND, PPOA, PPR, PPS, PR, PPOT, PINT, PAPASS, PREAL e PTRA

V – são as etiquetas referentes aos verbos – VAUX, VLIG, VINT, VTD, VTI e VBI

Loc – são as etiquetas referentes as locuções – LADV, LCONJ, LDEN, LPREP e LP

Cont – são as etiquetas referentes as contrações – PREP+ART, PREP+PREP, PREP+PD, PREP+PPR, PREP+PPOT, PREP+ADJ, PREP+N, PREP+PPOA, PREP+ADV, PPOA+PPOA, ADV+PPR, ADV+PPOA e ADJ+PPOA

Encl – são as etiquetas referentes as ênclices – VTD+PPOA, VTD+PAPASS, VAUX+PPOA, VBI+PPOA,

VLIG+PPOA, VTI+PPOA, VTI+PREAL, VBI+PPOA, VINT+PREAL, VINT+PPOA, VINT+PAPASS, VBI+PPR,

VTD+PPR, VTD+PREAL e VBI+PAPASS

Res – são as etiquetas – RES e IL

Pon – são as etiquetas referentes as pontuações

¹⁵ Etiqueta que não apareceu no corpus de teste.

PIND	83,87%	VTI+PPOA	0%
PPOA	81,39%	VTI+PREAL	-
PPR	93,75%	VINT+PREAL	-
PPS	98,27%	VINT+PPOA	0%
PR	81,34%	VINT+PAPASS	0%
PPOT	100%	VBI+PPR	-
PINT	0%	VTD+PPR	-
PAPASS	0%	VTD+PREAL	0%
PREAL	0%	VBI+PAPASS	-
PTRA	-	VAUX!PPOA	-
PREP	92,61%	VTD!PPOA	-
VAUX	79,66%	RES	15%
VLIG	84,71%	IL	-
VINT	57,39%	.	100%
VTD	76,40%	:	100%
VTI	34,43%	;	100%
VBI	23,25%	-	100%
I	-	(100%
LADV	53,66%	!	-
LCONJ	59,09%	?	100%
LPREP	70,83%	...	100%
LP	50%)	100%
LDEN	66,67%	"	100%
PDEN	-	[100%
PREP+ART	96,59%]	100%
PREP+PREP	-	{	-
PREP+PD	96,08%	}	-
PREP+PPR	85,71%	,	100%
PREP+PPOT	-	:	-
PREP+ADJ	0%		
PREP+N	100%		

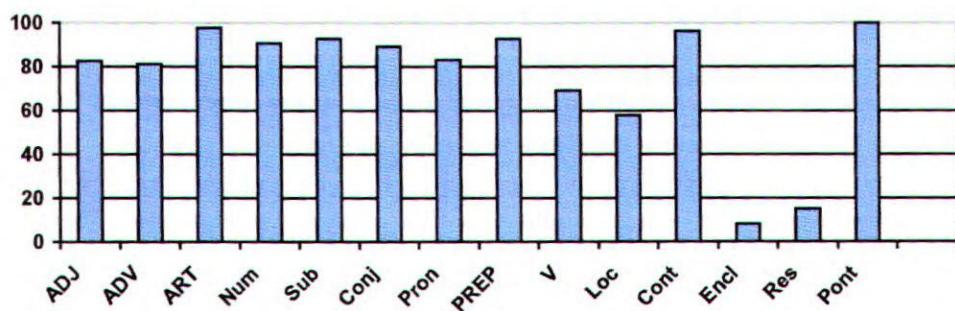


Figura 26 - Precisão por etiquetas em grupos - TreeTagger

4.2.1.2 Etiquetador baseado em transformação (TBL)

Nos experimentos com o etiquetador TBL foram mantidos todos os parâmetros default:

- Para o aprendizado das regras para palavras desconhecidas foi mantido o valor 300 – apenas bigramas em que pelo menos uma das palavras for uma das 300 mais frequentes serão utilizados na construção das regras.
- O programa de aprendizado de regras para palavras desconhecidas parará de rodar quando a pontuação da melhor regra encontrada cair abaixo do limiar igual a 3, valor default e que é indicado por Brill para textos com menos de 50K.
- A primeira decisão do etiquetador inicial será marcar as palavras que começam com letras maiúsculas como substantivo próprio e as que começam com minúsculas como substantivo comum, assim onde havia no código NNP¹⁶ foi alterado para NP e onde havia NN para N.
- O programa de aprendizado de regras contextuais parará de rodar quando a pontuação da melhor regra encontrada cair abaixo de um limiar igual a 2, valor default e que também é indicado por Brill para textos com menos de 50K.

O treinamento em uma Sun Ultra Interprise 3000 com 2 Gb de RAM foi bem mais lento do que o do TreeTagger como pode ser visto na Tabela 12. O modelo gerado por este treinamento é composto de X regras para tratar palavras desconhecidas e Y regras contextuais. A precisão geral deste etiquetador é um pouco melhor que a do etiquetador TreeTagger 88,76% (Tabela 13).

Tabela 12 - Tempo de treinamento e etiquetagem - TBL

	Tempo de treinamento – regras para palavras desconhecidas	Tempo de treinamento – regras contextuais	Tempo de etiquetagem
TBL (C3)	15 horas, 54 minutos e 33 segundos	6 horas, 14 minutos e 14 segundos	-
Corpus de calibração (C31)	-	-	7 segundos
Corpus de teste (C32)	-	-	6 segundos

Tabela 13 - Precisão Geral - TBL

	Precisão Geral
TBL (C3)	88.09%
Corpus de calibração (C31)	
Corpus de teste (C32)	88.76%

A Tabela 14 mostra as precisões por etiqueta do etiquetador TBL. A porcentagem de etiquetas problemáticas para o TBL é 33,34%, sendo que as etiquetas que são problemáticas para

o etiquetador TBL e TreeTagger estão em vermelho, enquanto as que são problemáticas apenas para o TBL estão em verde. Nota-se que uma das etiquetas — ORD — que não eram problemáticas para o TreeTagger é problemática para o TBL e que uma — VTD+PPOA — que era problemática para o TreeTagger não é para o TBL. A Figura 27 resume a Tabela 14.

Tabela 14 - Precisão por etiquetas no corpus de teste- TBL

Etiqetas	Precisão	Etiqetas	Precisão
ADJ	83,03%	PREP+PPOA	-
ADV	88,80%	PREP+ADV	100%
ART	98,63%	PPOA+PPOA	-
NC	96,99%	ADV+PPR	-
ORD	68,75 ^s	ADV+PPOA	0*
NO	50*	ADJ+PPOA	-
N	91,92%	VTD+PPOA	83,12
NP	95,97%	VTD+PAPASS	0*
CONCOORD	94,78%	VAUX+PPOA	100%
CONSUB	73,56 ^s	VBI+PPOA	35,71 ^s
PD	92,31%	VLIG+PPOA	50*
PIND	90,32%	VTI+PPOA	0*
PPOA	81,39%	VTI+PREAL	-
PPR	100%	VINT+PREAL	-
PPS	96,55%	VINT+PPOA	0*
PR	80,6%	VINT+PAPASS	0*
PPOT	100%	VBI+PPR	-
PINT	0*	VTD+PPR	-
PAPASS	0*	VTD+PREAL	0*
PREAL	0*	VBI+PAPASS	-
PTRA	-	VAUX! PPOA	-
PREP	93,64%	VTD! PPOA	-
VAUX	67,8 ^s	RES	40%
VLIG	84,08%	IL	-
VINT	47,83 ^s	:	100%
VTD	70,45%	;	100%
VTI	23,77%	-	100%
VBI	6,98 ^s		100%
I	-	(100%
LADV	43,90%	!	-
LCONJ	45,45%	?	100%
LPREP	58,33 ^s	...	100%
LP	0*)	100%
LDEN	50*	"	100%
PDEN	-	[100%
PREP+ART	97,73%]	100%
PREP+PREP	-	{	-
PREP+PD	94,12%	}	-
PREP+PPR	100%	,	100%
PREP+PPOT	-	'	-
PREP+ADJ	0*		
PREP+N	100%		

¹⁶ No conjunto de etiquetas utilizado por Brill, NNP é a etiqueta para substantivo próprio e NN a etiqueta para

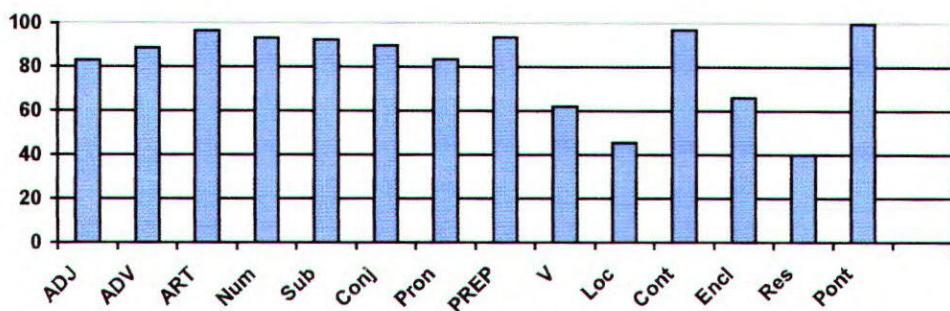


Figura 27 - Precisão por etiquetas em grupos - TBL

4.2.1.3 MXPOST

O etiquetador MXPOST não possui parâmetros para serem alterados. Foi o etiquetador que obteve maior precisão geral — 89,66% — como pode ser visto na Tabela 15. Os tempos de treinamento e teste são mostrados na Tabela 16.

Tabela 15 - Precisão Geral - MXPOST

	Precisão Geral
MXPOST (C3)	
Corpus de calibração (C31)	88,70%
Corpus de teste (C32)	89,66%

Tabela 16 - Tempo de treinamento e teste - MXPOST

	Tempo de treinamento	Tempo de etiquetagem
MXPOST (C3)	1 hora 28 minutos e 27 segundos	-
Corpus de calibração (C31)	-	3 minutos e 12 segundos
Corpus de teste (C32)	-	3 minutos e 14 segundos

A porcentagem de etiquetas problemáticas para o MXPOST é 38,46%. A Tabela 17 mostra as precisões por etiquetas, mostrando que todas as etiquetas que eram problemáticas para o TreeTagger e para o TBL são também problemáticas (em vermelho) para o MXPOST e que a que era apenas para o TBL também é para o MXPOST (em verde) e que uma das classes problemáticas para o TreeTagger que não era problemática para o TBL é para o MXPOST substantivo comum.

(VTD+PPOA) e que as classes – PPOA, PR e PREP+ADV que não eram problemáticas nem para o TreeTagger nem para o TBL são para o MXPOST (azul). Comparando as Figuras 28, 27 e 26 nota-se que os grupos de etiquetas problemáticas são os mesmos para os três etiquetadores, variando apenas a proporção.

Tabela 17 - Precisão por etiquetas no corpus de teste - MXPOST

Etiquetas	Precisão	Etiquetas	Precisão
ADJ	83,88%	PREP+PPOA	-
ADV	86,4%	PREP+ADV	50%
ART	98,88%	PPOA+PPOA	-
NC	95,49%	ADV+PPR	-
ORD	50%	ADV+PPOA	0%
NO	0%	ADJ+PPOA	-
N	94,61%	VTD+PPOA	72,73%
NP	96,7%	VTD+PAPASS	66,67%
CONCOORD	93,96%	VAUX+PPOA	100%
CONJSUB	74,53%	VBI+PPOA	28,57%
PD	92,31%	VLIG+PPOA	50%
PIND	80,64%	VTI+PPOA	0%
PPOA	74,42%	VTI+PREAL	-
PPR	100%	VINT+PREAL	-
PPS	96,55%	VINT+PPOA	0%
PR	77,61%	VINT+PAPASS	0%
PPOT	100%	VBI+PPR	-
PINT	0%	VTD+PPR	-
PAPASS	7,14%	VTD+PREAL	0%
PREAL	0%	VBI+PAPASS	-
PTRA	-	VAUX!PPOA	-
PREP	93,43%	VTD!PPOA	-
VAUX	66,95%	RES	45+
VLIG	84,08%	IL	-
VINT	53,04%	.	100%
VTD	72,61%	:	100%
VTI	39,34%	;	100%
VBI	32,56%	-	100%
I	-	(100%
LADV	43,9%	!	-
LCONJ	77,27%	?	100%
LPREP	58,33%	...	100%
LP	0%)	100%
LDEN	33,33%	"	100%
PDEN	-	[100%
PREP+ART	97,59%]	100%
PREP+PREP	-	{	-
PREP+PD	90,2%	}	-
PREP+PPR	100%	,	100%
PREP+PPOT	-	'	-
PREP+ADJ	0%		
PREP+N	100%		

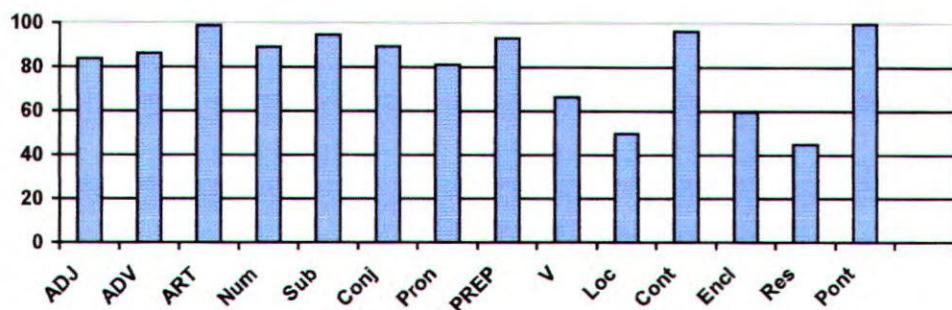


Figura 28 - Precisão por etiquetas em grupo - MXPOST

Por ser o etiquetador de maior precisão geral foi o etiquetador escolhido para verificarmos o impacto de um corpus de treinamento maior na precisão geral e também o impacto das palavras desconhecidas. Treinando o etiquetador com 90% do corpus (80% de treinamento + 10% de calibração) e testando com os 10% de teste foi encontrada uma precisão geral de 90,25%, ou seja, com 10% a mais de corpus de treinamento a precisão geral aumentou 0,59%. Treinando o etiquetador com 100% do corpus de treinamento e testando com os 10% do teste, ou seja, não existem palavras desconhecidas – foi obtida uma precisão de 93,65%.

4.2.1.4 PoSiTagger – Portuguese Simbolic Tagger

Sou definitivamente contra o definido, porque o definido é o bastante e o bastante não basta. – Fernando Pessoa

O etiquetador PoSiTagger é um etiquetador simbólico construído no Nilc, que teve suas regras construídas por uma lingüista, em vinte dias. As regras foram elaboradas tomando por base as regras de desambiguação morfossintática do ReGra (Apêndice C1) e as regras contextuais do etiquetador TBL (Apêndice C2). A elaboração de tais regras foi feita seguindo dicionários, dicionário inverso, gramáticas e vocabulário ortográfico da língua portuguesa (Pinheiro, 1990; Luft, 1993; Biderman, 1992; Lima, 1992; Kury, 1993; André, 1990; Fernandes, 1995; d'Andrade, 1993; Ibaixe, 1994; Cunha & Cintra, 1985; Faraco & Moura, 1994; Academia Brasileira de Letras, 1998).

A etiquetagem se dá em duas etapas: inicial e contextual. A etiquetagem incial é feita em duas fases: primeiro, o texto é etiquetado segundo uma lista de regras lexicalizadas preestabelida e depois, as palavras que não tiverem sido etiquetadas segundo estas regras e se estiverem presentes no léxico são etiquetadas com a etiqueta a elas associadas no léxico, caso não sejam palavras contempladas pelo léxico é utilizado o léxico de sufixos e prefixos¹⁷. Se ainda assim a palavra não for etiquetada é atribuída a ela a etiqueta RES caso inicie com letra minúscula, pressupondo-se que esta palavra se trata de uma palavra estrangeira, um coloquialismo, ou regionalismo, e caso tenha inicial maiúscula é etiquetada como substantivo próprio.

Na etiquetagem contextual, as etiquetas são alteradas de acordo com um contexto variável que pode ser formado por: palavras anteriores àquela que está sendo etiquetada, posteriores, etiquetas anteriores e posteriores; e pelos atributos: a existência ou não de uma dada palavra ou etiqueta no período, o fato de a palavra ser a primeira do período ou não, qual é a primeira palavra do período, qual é a última palavra do período ou qualquer combinação destas características utilizando-se de operadores e, ou e não. A Figura 29 mostra a arquitetura deste etiquetador.

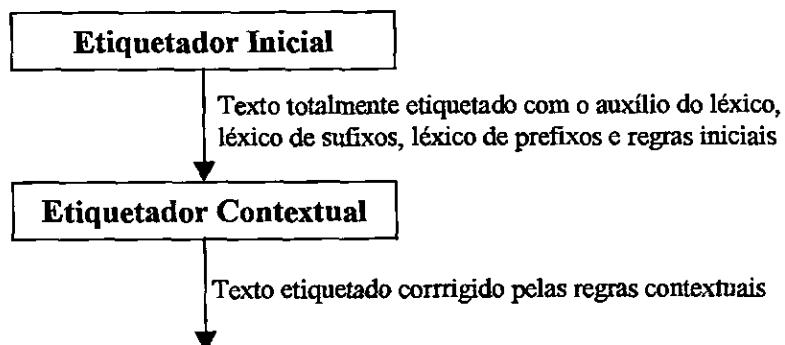


Figura 29 - Arquitetura do PoSiTagger

Apenas dois ítems são obrigatórios nas regras: a palavra e/ou etiqueta que deve ser alterada e a nova etiqueta que deve ser utilizada. As regras iniciais e as regras contextuais não estão no código do etiquetador mas em arquivos a parte para facilitar futuras alterações. Nestes experimentos o contexto utilizado foi de até sete palavras/etiquetas a esquerda/direita da palavra em foco, mas pode ser alterado para qualquer valor desejado. As regras utilizadas neste experimento estão no Apêndice C3.

¹⁷ Prefixos e sufixos são adicionados e retirados da palavra tentando encontrar uma palavra conhecida pelo léxico.

A precisão geral é mostrada na Tabela 18. Os tempos gastos na etiquetagem são mostrados na Tabela 19. A tabela 20 mostra as precisões por etiqueta e a Figura 31 mostra as informações da Tabela 20 agrupando as etiquetas.

Tabela 18 - Precisão Geral do PoSiTagger

PoSiTagger	Precisão Geral
Corpus de calibração (C31)	83,57%
Corpus de teste (C32)	82,65%

Tabela 19 - Tempos de etiquetagem do PoSiTagger

PoSiTagger	Tempo de etiquetagem
Corpus de calibração (C31)	25 minutos e 41 segundos
Corpus de teste (C32)	28 minutos e três segundos

Tabela 20 - Precisão Geral do PoSiTagger nos corpus de teste e calibração

Etiquetas	Precisão	Etiquetas	Precisão
ADJ	88,26%	PREP+PPOA	-
ADV	44%	PREP+ADV	50%
ART	97,25%	PPOA+PPOA	-
NC	100%	ADV+PPR	-
ORD	68,75%	ADJ+PPOA	-
NO	100%	VTD+PPOA	37,66%
N	85,02%	VTD+PAPASS	100%
NP	73,26%	VAUX+PPOA	0%
CONCOORD	90,38%	VBI+PPOA	7,14%
CONJSUB	56,60%	VLIG+PPOA	0%
PD	38,46%	VTI+PPOA	14,28%
PIND	61,29%	VTI+PREAL	-
PPOA	27,90%	VINT+PREAL	-
PPR	93,75%	VINT+PPOA	0%
PFS	18,96%	VINT+PAPASS	0%
PR	72,39%	VBI+PAPASS	-
PPOT	100%	VBI+PPR	-
PINT	100%	VTD+PPR	-
PAPASS	0%	VTD+PREAL	0%
PREAL	0%	VBI+PAPASS	-
PTRA	-	VAUX!PPOA	-
PREP	87,49%	VTD!PPOA	-
VAUX	38,98%	RES	97,5%
VLIG	91,08	IL	-
VINT	60%	:	100%
VTD	62,16%	;	100%
VTI	52,46%		100%

VBI	44,19%	-	100%
I	-	(0%
LADV	65,85%	!	-
LCONJ	68,18%	?	100%
LPREP	83,33%	...	100%
LP	100%)	100%
LDEN	50%	"	100%
PDEN	-	[100%
PREP+ART	93,19%]	100%
PREP+PREP	-	{	-
PREP+PD	56,86%	}	-
PREP+PPR	100%	,	99,76%
PREP+PPOT	-	:	-
PREP+ADJ	100%		
PREP+N	100%		

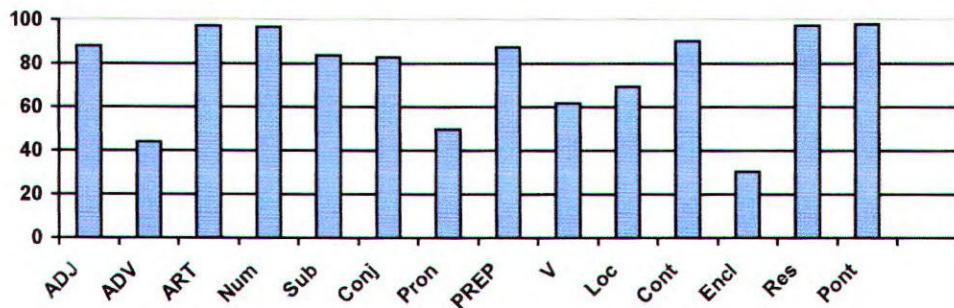


Figura 30 - Precisão por grupos de etiquetas - PoSiTagger

A Tabela 20 mostra que oito das etiquetas (em cor-de-rosa) que não eram problemáticas para nenhum dos três etiquetadores das seções anteriores são para o PoSiTagger. Em contrapartida mostra que sete das etiquetas que são problemáticas — NO, PINT, LPREP, LP, PREP+ADJ, VTD+PAPASS e RES — para os três etiquetadores não são pra ele e que para quatro destas ele tem 100% de precisão. Fora estas sete etiquetas todas as outras que eram problemáticas para um dos etiquetadores ou para todos eles, o são também para o PoSiTagger. A Figura 31 contrapõe os resultados dos quatro etiquetadores por grupos de etiquetas.

¹⁸ O erro na etiquetagem deste símbolo foi causado por não ter sido colocado na lista de símbolos de pontuação, tendo sido etiquetado pelo PoSiTagger como residual.

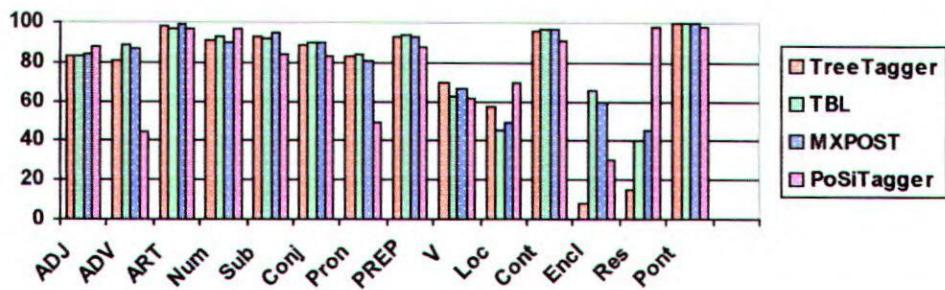


Figura 31 - Precisão por etiqueta dos etiquetadores TreeTagger, TBL, MXPOST e PoSiTagger

4.2.2 Avaliação do conjunto de etiquetas

Aprenda cada qual a caminhar pela estrada que mais lhe convenha - Propércio

Para avaliar qual o grau de dificuldade imposto por termos em nosso conjunto de etiquetas as que consideram a transitividade, treinamos o etiquetador de maior precisão - MXPOST – com um conjunto de etiquetas em que a única diferença com relação ao NILC tagset é que verbos não são subcategorizados. A precisão obtida treinando o etiquetador com os mesmos 80% dos dados utilizados anteriormente foi 92,51%, 2,85% menos erros no teste.

4.2.3 Avaliação dos tipos de texto

Quanto aos tipos de texto utilizados no treinamento e teste, havia duas perguntas a serem respondidas:

- 1) Se existem métodos de etiquetagem que tem uma maior precisão com um certo tipo de texto
- 2) Qual o impacto do uso de tipos de texto no teste que são diferentes do treinamento

Para responder a primeira pergunta foram feitos 3 treinamentos com os etiquetadores TreeTagger como trígrama e unígrama, TBL e MXPOST, utilizando em cada um deles um dos três conjuntos de textos – didático, literário e jornalístico (C4, C5 e C6) e etiquetagens com os conjuntos de texto de calibração e teste – didático, literário e jornalístico (C41, C42, C51, C52,

C61 e C62). Também foram etiquetados os corpus C41, C42, C51, C52, C61 e C62 com o PoSiTagger. A Tabela 21 mostra as precisões de cada um dos etiquetadores para textos didáticos, a Tabela 22 para jornalísticos e a Tabela 23 para literários.

Tabela 21 - Precisão geral dos etiquetadores para textos didáticos

Etiquetadores	Precisão Geral
<i>TreeTagger – Unigrama (C4)</i>	
Conjunto de textos didáticos de calibração (C41)	83,30%
Conjunto de textos didáticos de teste (C42)	82,50%
<i>TreeTagger – Trigrama (C4)</i>	
Conjunto de textos didáticos de calibração (C41)	89,11%
Conjunto de textos didáticos de teste (C42)	86,64%
<i>TBL (C4)</i>	
Conjunto de textos didáticos de calibração (C41)	87,89%
Conjunto de textos didáticos de teste (C42)	88,19%
<i>MXPOST (C4)</i>	
Conjunto de textos didáticos de calibração (C41)	89,05%
Conjunto de textos didáticos de teste (C42)	88%
<i>PoSiTagger (C4)</i>	
Conjunto de textos didáticos de calibração (C41)	81,65
Conjunto de textos didáticos de teste (C42)	83,24%

Tabela 22 - Precisão geral dos etiquetadores para textos jornalísticos

Etiquetadores	Precisão Geral
<i>TreeTagger – Unigrama (C5)</i>	
Conjunto de textos jornalísticos de calibração (C51)	79,33%
Conjunto de textos jornalísticos de teste (C52)	79,73%
<i>TreeTagger – Trigrama (C5)</i>	
Conjunto de textos jornalísticos de calibração (C51)	88,32
Conjunto de textos jornalísticos de teste (C52)	88,9
<i>TBL (C5)</i>	
Conjunto de textos jornalísticos de calibração (C51)	88,57
Conjunto de textos jornalísticos de teste (C52)	88,53
<i>MXPOST (C5)</i>	
Conjunto de textos jornalísticos de calibração (C51)	88,68
Conjunto de textos jornalísticos de teste (C52)	88,67
<i>PoSiTagger (C5)</i>	
Conjunto de textos jornalísticos de calibração (C51)	83,51
Conjunto de textos jornalísticos de teste (C52)	81,59

Tabela 23 - Precisão geral dos etiquetadores para textos literários

Etiquetadores	Precisão Geral
<i>TreeTagger – Unigrama (C6)</i>	
Conjunto de textos literários de calibração (C61)	73,53%
Conjunto de textos literários de teste (C62)	74,32%
<i>TreeTagger – Trigrama (C6)</i>	
Conjunto de textos literários de calibração (C61)	82,46
Conjunto de textos literários de teste (C62)	84,08
<i>TBL (C6)</i>	

Conjunto de textos literários de calibração (C61)	83,33
Conjunto de textos literários de teste (C62)	85,35
<i>MXPOST (C6)</i>	
Conjunto de textos literários de calibração (C61)	83,11%
Conjunto de textos literários de teste (C62)	86,85%
<i>PoSITagger (C6)</i>	
Conjunto de textos literários de calibração (C61)	84,30%
Conjunto de textos literários de teste (C62)	84,23%

Analizando as Tabelas 21, 22 e 23 poderia se concluir com os resultados para o conjunto de textos de teste que o etiquetador TBL é melhor para textos didáticos, que o etiquetador TreeTagger como trígrama é melhor para textos jornalísticos e que o etiquetador MXPOST é o melhor para textos literários. No entanto, no conjunto de textos de calibração o etiquetador TreeTagger é o melhor para textos didáticos, o MXPOST para textos jornalísticos e o TBL para textos literários. O que mostra que não existe uma relação entre tipo de texto e etiquetador ou, o mais provável, que apenas um teste não é capaz de verificar e determinar a relação existente entre estes textos, ou ainda indicar que existem problemas em nossos dados de treinamento e que alguns dos etiquetadores conseguem obter melhores resultados que outros frente a dados com alguma inconsistência. Este teste serviu também para comprovar o grau de dificuldade imposto por cada tipo de texto. Note que a pior média dos resultados é a dos dos etiquetadores literários, melhor um pouco se saem os jornalísticos e a melhor, apesar de serem os de menor corpus de treinamento, é a dos didáticos.

Para responder a segunda pergunta testamos cada um dos etiquetadores treinados para um único tipo de texto com um conjunto de teste formado pela união proporcional dos conjuntos de teste dos outros dois tipos. As Tabelas 24, 25 e 26 mostram que uma vez treinado com um tipo de texto um etiquetador não se sai bem quando testado com outros tipos de texto. O que indica que existem duas opções para etiquetar textos dos três graus de escrita, citados na Seção 4.1.1:

- ter um corpus de treinamento equilibrado com os diversos tipos de texto de interesse, contendo um grande número de exemplos para que o etiquetador generalize corretamente;
- ter um conjunto de etiquetadores especialistas, treinados cada um com um único tipo de texto de forma que um conjunto de exemplos pequeno e cheio de exceções não prejudique o aprendizado do etiquetador.

Tabela 24 - Etiquetadores didáticos frente a textos jornalísticos e literários

Etiquetadores	Precisão Geral
<i>TreeTagger – Unigrama (C4)</i> Conjuntos de textos de 2 gêneros de calibração JL (C7) Conjuntos de textos de 2 gêneros de teste JL (C8)	64,93% 68,80%
<i>TreeTagger – Trigrama (C4)</i> Conjuntos de textos de 2 gêneros de calibração JL (C7) Conjuntos de textos de 2 gêneros de teste JL (C8)	74,74% 81,63%
<i>TBL (C4)</i> Conjuntos de textos de 2 gêneros de calibração JL (C7) Conjuntos de textos de 2 gêneros de teste JL (C8)	77,84% 81,07%
<i>MXPOST (C4)</i> Conjuntos de textos de 2 gêneros de calibração JL (C7) Conjuntos de textos de 2 gêneros de teste JL (C8)	77,65% 79,39%

Tabela 25 - Etiquetadores jornalísticos frente a textos didáticos e literários

Etiquetadores	Precisão Geral
<i>TreeTagger – Unigrama (C4)</i> Conjuntos de textos de 2 gêneros de calibração DL (C9) Conjuntos de textos de 2 gêneros de teste DL (C10)	73,98% 73,54%
<i>TreeTagger – Trigrama (C4)</i> Conjuntos de textos de 2 gêneros de calibração DL (C9) Conjuntos de textos de 2 gêneros de teste DL (C10)	82,12% 82,1%
<i>TBL (C4)</i> Conjuntos de textos de 2 gêneros de calibração DL (C9) Conjuntos de textos de 2 gêneros de teste DL (C10)	82,66% 84,02%
<i>MXPOST (C4)</i> Conjuntos de textos de 2 gêneros de calibração DL (C9) Conjuntos de textos de 2 gêneros de teste DL (C10)	82,58% 84,64%

Tabela 26 - Etiquetadores literários frente a textos didáticos e jornalísticos

Etiquetadores	Precisão Geral
<i>TreeTagger – Unigrama (C4)</i> Conjuntos de textos de 2 gêneros de calibração DJ (C11) Conjuntos de textos de 2 gêneros de teste DJ (C12)	73,28% 73,38%
<i>TreeTagger – Trigrama (C4)</i> Conjuntos de textos de 2 gêneros de calibração DJ (C11) Conjuntos de textos de 2 gêneros de teste DJ (C12)	81,56% 80,49%
<i>TBL (C4)</i> Conjuntos de textos de 2 gêneros de calibração DJ (C11) Conjuntos de textos de 2 gêneros de teste DJ (C12)	81,72% 81,49%
<i>MXPOST (C4)</i> Conjuntos de textos de 2 gêneros de calibração DJ (C11) Conjuntos de textos de 2 gêneros de teste DJ (C12)	83,02% 81,87%

5 COMBINAÇÃO DE ETIQUETADORES

A união faz a força – ditado popular

O maior objetivo de um classificador é predizer com sucesso novos casos apresentados. No entanto, para a maioria dos problemas tratados a quantidade de dados disponíveis é limitada e muitas vezes traz informações superficiais. Este limite quanto ao volume e qualidade dos dados torna bem mais difícil a tarefa de classificar corretamente um dado não visto no treinamento e foi para superá-lo que surgiu em Aprendizado de Máquina (AM) a idéia de combinar classificadores.

Se diferentes classificadores erram de forma diferente, ou por serem baseados em diferentes formalismos ou por conterem conhecimento diferente, então combinar um conjunto de classificadores (*ensemble*) é uma forma de minimizar os erros (aumentar a precisão geral), explorando as situações em que um classificador erra e outro acerta. A combinação de classificadores surge como uma técnica para ser utilizada para resolver problemas mais gerais que envolvem diferentes tipos de dados ou para ser aplicada em problemas que não foram resolvidos satisfatoriamente por um único classificador e/ou em problemas cujo desempenho deve ser estável, isto é, termos um classificador robusto (para uma revisão da área veja (Dietterich, 1997; Chan et al., 1999)).

A combinação de classificadores tem sido utilizada em diversas áreas em Inteligência Artificial, como Redes Neurais (Sharkey, 1999), Aprendizado Simbólico de Máquina (Barnett, 1981), Engenharia de Software (Knight, 1986) e mais recentemente em PLN em tarefas como

dizambigüização semântica (Rigau et al., 1997), avaliação de heurísticas (Agirre et al., 1998) e etiquetagem morfossintática (van Halteren et al., 1998; Brill & Wu, 1998), mostrando resultados melhores que o do melhor dos classificadores individuais quando os erros dos classificadores não estão relacionados.

O método de combinação escolhido deveria, então, aproveitar os pontos fortes de classificadores individuais, evitar os seus pontos fracos, melhorando a precisão da classificação. Entretanto, é difícil combinar as saídas de tal forma que as falhas de alguns não interfiram nas saídas corretas de outros. Esta é uma situação ideal, porém não existe nenhum método que a garanta. Segundo Sharkey (1996), o método de combinação envolveria três aspectos:

- A escolha de quantos classificadores serão combinados.
- A criação ou seleção do conjunto de classificadores para serem combinados.
- O método pelo qual as saídas dos classificadores escolhidos serão combinadas.

A escolha de quantos classificadores serão utilizados depende do problema em questão e do número de classes utilizadas. Há apenas uma sugestão comum na literatura quanto a este critério, que é a sugestão intuitiva de usar um número de classificadores ímpar, de preferência um número primo, para que fique mais difícil a ocorrência de conflitos, em que duas classes distintas possuem o mesmo número de votos para serem selecionadas como a classe associada a um novo padrão.

O conjunto de classificadores pode ser formado treinando diferentes algoritmos com um único conjunto de dados de treinamento, ou treinando o mesmo algoritmo com diferentes conjuntos de dados de treinamento ou com diferentes partes de um mesmo conjunto de dados. Dentro os classificadores formados devem ser utilizados os que não produzem muitos erros semelhantes. Neste trabalho, para selecionar quais etiquetadores seriam combinados utilizamos a mesma medida utilizada por Brill & Wu (Brill & Wu, 1998) que será mostrada na Seção 5.1.

As saídas dos classificadores podem ser combinadas de três formas: em paralelo, em cascata e de forma hierárquica. No modelo em paralelo (ou cooperativo), os classificadores possuem o mesmo conjunto de entradas e produzem saídas pertencentes ao mesmo conjunto de possíveis classes, cooperando para classificar um padrão de entrada (Figura 32). No modelo em cascata (ou sucessivo), a saída de um classificador serve como entrada para outro classificador (Figura 33). Já o modelo hierárquico mistura os modelos paralelo e cascata, em que os classificadores trabalham cooperativamente, mas há um classificador trabalhando em cascata,

fazendo o papel de supervisor, fornecendo dados ou pesos para um ou mais dos classificadores que estão trabalhando em paralelo (refs) (Figura 34).

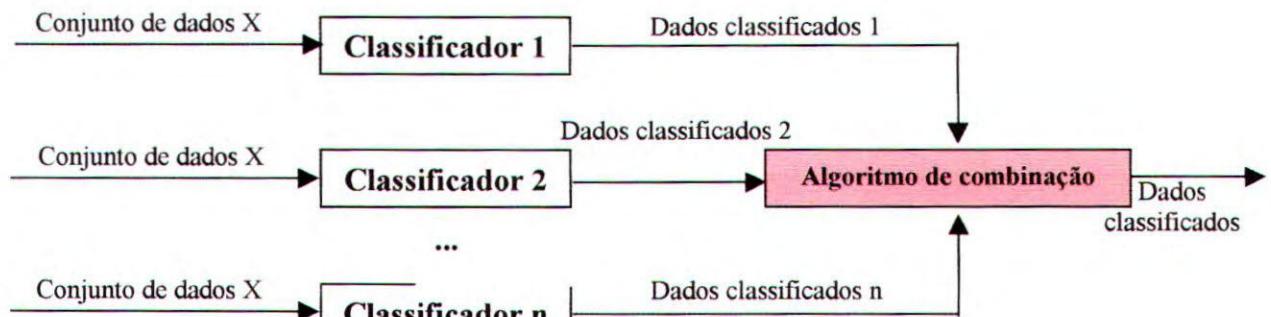


Figura 32 - Combinação de classificadores em paralelo

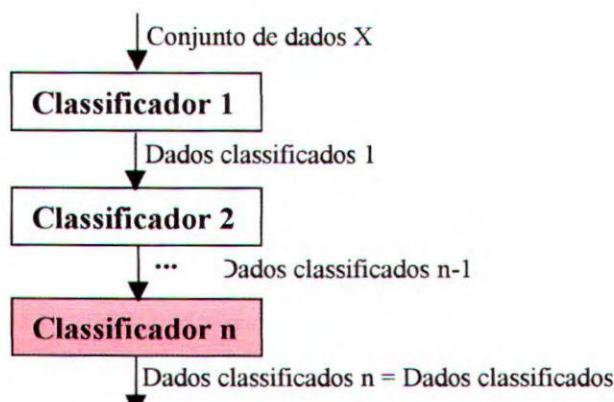


Figura 33 - Combinação de classificadores em cascata

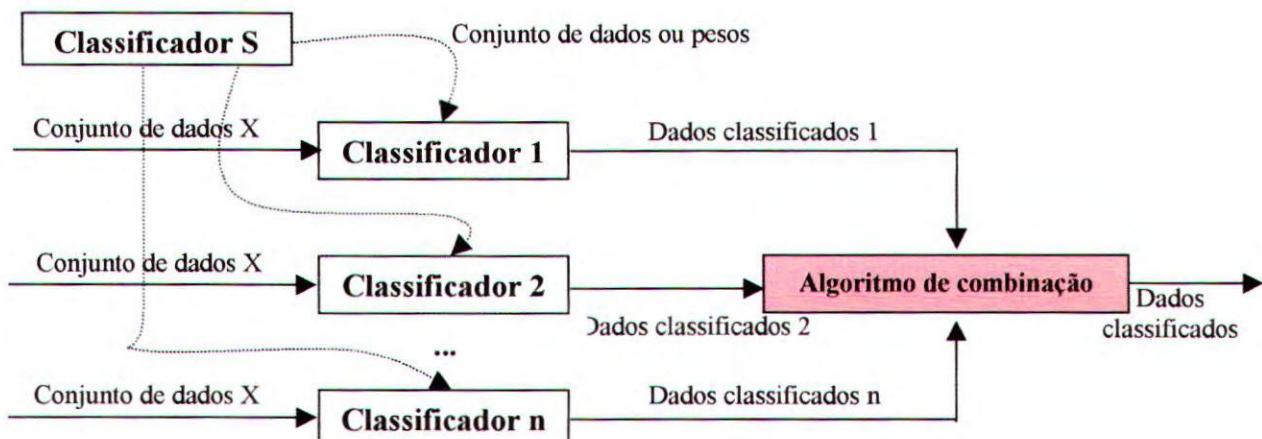


Figura 34 - Combinação hierárquica de classificadores

Fazendo uma analogia entre as formas de combinar os classificadores e um jogo de futebol, o modelo em cascata seria o processo de marcar um gol, o modelo em paralelo seriam os comentaristas decidirem se uma jogada foi ou não falta e o modelo hierárquico seria a estratégia dos jogadores em campo. Para o exemplo do modelo em cascata, o time de futebol seria o conjunto de classificadores com os classificadores especialistas — goleiro, zagueiros, laterais, meio-campo, pontas-de-lança e atacantes —, e o objetivo seria marcar gols. Cada jogador faria o seu papel de especialista, o goleiro defenderia e faria um arremesso, o meio-campo passaria a bola para o ponta-de-lança que driblaria um jogador do time adversário e faria um lançamento para um atacante que marcaria o gol. A bola seria como o dado a ser classificado e cada ação de cada jogador por onde ela passou seria como a sua classificação para a entrada bola. A classificação definitiva seria a dada pelo último classificador da sequência, o atacante. Para o modelo em paralelo, o conjunto de classificadores seriam os comentaristas, o dado seria a jogada e cada um deles, dado seu ângulo, ponto de vista, e conhecimento dariam seu voto dizendo se foi ou não falta, o sistema de decisão seria o narrador que diria se realmente foi ou não falta, dados o número de votos a favor e contra e/ou a confiança que ele tem em cada um dos comentaristas. No modelo hierárquico, o time seria mais uma vez o conjunto de classificadores, cada jogador decide qual a estratégia seguir durante o jogo, mas há uma interferência de um supervisor na decisão de um ou mais jogadores. Este supervisor poderia ter conhecimento prévio das principais especialidades de cada jogador (técnico) ou poderia ser um dos classificadores e estar avaliando isto durante o jogo (capitão do time). Neste modelo, a decisão final sai das decisões de todos os jogadores, mas o processo é influenciado pelo supervisor.

A escolha de qual forma utilizar depende do problema, não havendo recomendações para tal escolha, de forma que deve-se estudar mais de um método e escolher o mais rápido e eficiente para o problema em questão.

Neste trabalho, foram utilizados o modelo em cascata e o modelo em paralelo. As formas utilizadas em nossos experimentos e os resultados são mostrados na Seção 5.2.

5.1 Taxa de complementaridade de Etiquetadores

Como falado anteriormente, a combinação de classificadores é viável quando existe um número razoável de situações em que um classificador erra e outro acerta, isto é, se são complementares. Para verificar quais os etiquetadores tratados neste trabalho eram complementares foi utilizada a mesma medida usada por Brill & Wu (1998) para analisar quão diferentes são os erros dos etiquetadores. A taxa de complementaridade dos etiquetadores X e Y é definida como:

$$\text{Comp}(X,Y) = (1 - (\# \text{ de erros comuns} / \# \text{ número de erros somente de } X)) * 100$$

e mede a porcentagem de vezes em que X está errado e Y correto. Como mostra a Tabela 27, as taxas são altas. Por exemplo, quando o MXPOST está errado, o TreeTagger como Trígrama está certo em 35,24% das vezes, e quando o TreeTagger como Trígrama está errado, MXPOST está certo em 38,99% das vezes.

Tabela 27 - Taxa de Complementaridade entre Etiquetadores

X	Y	TreeTagger como Unígrama	TreeTagger como Trígrama	TBL	MXPOST
TreeTagger como Unígrama	—	53.78%	60.84	64.48	
TreeTagger como Trígrama	21.44%	—	37.01	38.99	
TBL	32.96	36.56	—	34.32	
MXPOST	35.92	35.24	30.77	—	

Para a combinação gerar bons resultados, os etiquetadores escolhidos precisam cometer erros diferentes. A Tabela 28 mostra que, para 95,68% das palavras do corpus de teste, ao menos um dos quatro etiquetadores acerta ao etiquetar a palavra. Quando utilizamos apenas três

etiquetadores (não utilizamos o TreeTagger como unigrama), para 94,8% das palavras pelo menos um dos etiquetadores acertou (terceira coluna).

Tabela 28 – Concordância entre etiquetadores no teste

	Todos os etiquetadores	Etiquetadores sem o TreeTagger como unigrama
Todos os etiquetadores estão corretos	73,32%	0,03
Maioria correto	14,55	8,29
Correto presente sem maioria	4,25	83,20
Minoria correta	3,56	3,27
Todos os etiquetadores estão errados	4,32%	5,20

Em outras palavras, se tivéssemos um oráculo que pudesse sempre pegar a etiqueta correta das diferentes saídas, poderíamos ter um resultado melhor do que o resultado do melhor etiquetador. A Tabela 29 mostra que as taxas de erro diminuem à medida que acrescentamos mais etiquetadores na combinação. Os resultados da Tabela 29 são referentes ao maior acréscimo possível em relação à precisão do MXPOST. Por exemplo, quando o oráculo usa os quatro etiquetadores treinados, a taxa de erro é 4,32% o que significa uma redução de 58,22% sobre a taxa de erro do MXPOST. Podemos esperar melhorias aditivas apesar de não sabermos quando a saturação será alcançada.

Tabela 29 - Combinação Ideal

	MXPOST	+ Trígrama	+ TBL	+ Unigrama
% de vezes em que todos estão errados	10,34%	6,80%	5,20%	4,32%
% melhoria do oráculo	-	34,23%	49,71%	58,22%

5.2 Métodos para Combinação de Etiquetadores

Da discussão nasce a luz. - Provérbio

Neste trabalho, foram utilizadas três estratégias para combinar os etiquetadores baseadas nas variações do número de etiquetadores, da forma de gerar o conjunto de etiquetadores e do modelo de combinação das saídas dos etiquetadores para combinar os etiquetadores que são mostradas em detalhes nas próximas subseções.

5.2.1 Métodos paralelos baseados em diferentes algoritmos com um único conjunto de dados de treinamento

No modelo em paralelo, são três os principais métodos que podem ser utilizados para combinar as saídas: votação pela maioria simples (*simple voting*), votação ponderada (*weighted voting*) e *stacking*. No método votação pela maioria simples, cada classificador vota e o voto da maioria é a classe escolhida. Já no método votação ponderada o voto de cada classificador é pesado por sua precisão (Golding & Roth, 1999) ou usando métodos mais sofisticados, como por exemplo o uso do algoritmo *Naïve Bayes* para aprender pesos para os classificadores (Ali & Pazzani, 1996). Em *stacking*, um classificador (nível 1) é treinado para predizer a classe correta quando é fornecido como entrada as saídas de outros classificadores e informações adicionais (retiradas do corpus de calibração). *Stacking* não necessariamente seleciona uma das classes sugeridas pelos classificadores de nível 0. Neste trabalho, o classificador de nível 1 é baseado em casos. A Figura 35 mostra as três principais formas de combinar as saídas e as técnicas implementadas por van Halteren et al. (H) e por Brill & Wu (B) para combinação de etiquetadores em paralelo.

Decisão Aleatória ¹⁹	
Votação pela Maioria	Majority1(H) Majority2(B)
Votação Ponderada	Tot Precision (H) Tag Precision (H) Precision-Recall (H)
<i>Stacking</i>	TagPair (H) Tags (H) Tags+Word (H) Tags+Context (H) Pick Tag (B) Pick Tagger (B)

Figura 35 - Métodos para combinar as saídas em um modelo paralelo

Na Decisão Aleatória, dadas as saídas de todos os classificadores, a saída de um deles é escolhida aleatoriamente. Em Majority1, cada etiquetador vota e a etiqueta com maior número de votos é a escolhida e, em caso de empate, escolhe-se aleatoriamente entre as etiquetas vencedoras.

¹⁹ A decisão aleatória é utilizada apenas como base para comparação dos resultados.

Em Majority2, a etiqueta com maior número de votos é a escolhida e, em caso de empate, a etiqueta do etiquetador de maior precisão é a escolhida. Nós implementamos outra estratégia de votação pela maioria (Majority3), na qual a maioria é somente considerada quando pelo menos 51% dos etiquetadores votam na mesma etiqueta, e em caso de empate, a etiqueta do etiquetador de maior precisão é a escolhida.

Em Tot Precision, cada etiquetador vota sua precisão sobre a etiqueta sugerida, sendo adicionados os votos para a mesma classe, e o resultado final é a classe com mais votos. Em caso de empate, escolhe-se aleatoriamente entre as etiquetas vencedoras. Tag Precision segue a mesma idéia de Tot Precision, mas cada etiquetador vota sua precisão para a classe sugerida (*Precision*). Precision-Recall é semelhante ao último, mas cada etiquetador vota a sua precisão para a classe sugerida mais 1 menos o recall dos oponentes para a etiqueta que ele sugeriu. Os métodos de votação simples e ponderada são melhor detalhados na Figura 36.

Dado que E_i são os etiquetadores a ser combinados, que $e_i(ps)$ é a etiqueta mais votada²⁰ para uma palavra/símbolo ps como sugerido por E_i , e que a qualidade de um etiquetador é medida por:

- Precisão de E_i para a etiqueta et : $Prec(E_i, et)$
- Recall de E_i para a etiqueta et : $Rec(E_i, et)$
- Precisão geral de E_i : $Prec(E_i)$

Então o voto $V(et, ps)$ para a etiquetar a palavra/símbolo ps com a etiqueta et é dado por:

- **Majority1:**

$$\sum_i \text{Se } e_i(ps) = et \text{ ENTÃO et SENÃO 0}$$

- **Majority2:**

$$\sum_i \text{Se } e_i(ps) = et \text{ ENTÃO et SENÃO } Prec(E_i)$$

- **Majority3:**

$$\sum_i \text{Se } e_i(ps) = et \text{ ENTÃO et SENÃO } Prec(E_i)$$

- **Tot Precision:**

$$\sum_i \text{Se } e_i(ps) = et \text{ ENTÃO } Prec(E_i) \text{ SENÃO 0}^{21}$$

- **Tag Precision:**

$$\sum_i \text{Se } e_i(ps) = et \text{ ENTÃO } Prec(E_i, et) \text{ SENÃO 0}$$

- **Precision-Recall:**

$$\sum_i \text{Se } e_i(ps) = et \text{ ENTÃO } Prec(E_i, et) \text{ SENÃO } 1 - Rec(e_i, et)$$

Figura 36 - Algoritmos de votação simples e ponderada para a etiquetagem morfossintática

²⁰ Para Majority 3 a etiqueta mais votada é a que obteve mais de 50% dos votos.

TagPair permite que uma etiqueta sugerida pela minoria (ou nenhum) dos etiquetadores tenha chance de ~~vencer~~ usando informações dos etiquetadores em pares. Ele considera situações nas quais um etiquetador sugere T_1 e outro T_2 , e estima a probabilidade de nesta situação a etiqueta ser realmente T_x . Na combinação, cada par é selecionado e vota (com as probabilidades para a etiqueta apropriada) para cada etiqueta possível, não apenas as sugeridas pelos etiquetadores. Por exemplo, para os casos listados abaixo, a escolha seria T_{x2} .

T_1	T_2	T_x	Probabilidades
T_1	T_2	T_{x1}	(2/12)%
T_1	T_2	T_{x2}	(5/12)%
T_1	T_2	T_{x3}	(3/12)%
T_1	T_2	T_{x4}	(2/12)%

Entretanto, se um par de etiquetas (T_1-T_2) não foi observado no corpus de calibração, é usada informação de cada etiquetador em particular (a probabilidade de cada etiqueta T_x dado que o etiquetador sugeriu T_i).

Em Tags, ~~cada caso~~ consiste das etiquetas sugeridas por cada um dos etiquetadores e da etiqueta correta, enquanto que em Tags+Word a palavra em foco também é considerada. Durante a etiquetação a medida de similaridade usada é a freqüência da etiqueta. Em Tags+Context²¹, ~~cada~~ caso consiste da informação usada em Tags+Word mais as etiquetas sugeridas pelos ~~demais~~ etiquetadores para as palavras anterior e posterior. Em Pick Tag e Pick Tagger cada caso consiste de toda a informação de Tags+Context mais as palavras anterior e posterior. O primeiro usa este contexto para especificar que etiqueta escolher enquanto que o segundo usa para especificar em que etiquetador confiar.

Tendo como base os resultados apresentados na Seção 5, foi tomada a decisão de utilizar os quatro etiquetadores – TreeTagger como unígrama e como trígrama, TBL e MXPOST, treinados com o ~~corpus~~ de treinamento (C3) – na combinação em paralelo com as técnicas mostradas na Figura 35. Analisando a Tabela 30, vê-se que seis métodos superaram o melhor dos algoritmos individualmente e que a técnica de melhor resultado foi a TagPair com 90,91% de precisão. A Tabela 31, mostra em mais detalhes os resultados da técnica TagPair.

²¹ SENÃO 0 significa que em caso de empate a etiqueta é escolhida aleatoriamente entre as etiquetas votadas

²² Não foi implementado neste trabalho.

Tabela 30 - Resultados da combinação

Métodos	Precisão
Decisão Aleatória	86.43%
Majority1	89.16%
Majority2	89.20%
Majority3	90.41%
Tot Precision	90.70%
Tag Precision	90.72%
Precision-Recall	90.3%
TagPair	90.91%
Tags	89.66%
Tags+Word	90.04%
Pick Tag	88.51%
Pick Tagger	87.83%

Tabela 31 Resultados da combinação utilizando o método TagPair

Etiquetadores	% no corpus de teste	Média da % individual dos etiquetadores + % de melhora	Redução do erro quando comparando com o melhor
U ²³	80,01%	—	—
T	88,47%	—	—
B	88,76%	—	—
M	89,66%	—	—
UT	88,74%	84,24 + 4,5	* ²⁴
UB	89,49%	84,38 + 5,11	*
UM	90,04%	84,83 + 5,21	3,67%
TB	89,57%	88,61 + 0,96	*
TM	89,51%	89,06 + 0,45	*
BM	89,80%	89,21 + 0,59	1,35%
UTB	89,63%	85,75 + 3,88	*
UTM	89,73%	86,05 + 3,68	0,68%
UBM	90,61%	86,14 + 4,47	6,47%
TBM	90,91%	88,96 + 1,95	12,09%
UTBM	90,53%	86,72 + 3,81	8,41%

5.2.2 Métodos paralelos baseados em um único etiquetador

Nós somos aquilo que fazemos repetidamente. Excelência, então, não é um modo de agir, mas um hábito. -Aristóteles

Uma alternativa aos métodos apresentados na Seção 5.2.1 é fazer o treinamento com um único algoritmo, utilizando n partes do corpus de treinamento como entrada do algoritmo. Isto pode ser feito de duas maneiras:

²³ A letra "U" é utilizada para representar o TreeTagger como unígrafo, a letra "T" para representar o TreeTagger como trígrafo, a letra "B" é utilizada para representar o etiquetador TBL e a letra "M" para representar o etiquetador MXPOST.

²⁴ * Quando não houve redução de erro

- I) Dividindo-se o corpus de treinamento de acordo com os diversos tipos de informação nele contido – por exemplo, para o nosso caso, em textos didáticos, jornalísticos e literários. Em que para cada divisão é feito um treinamento.
- 2) Gerando n corpus de treinamento a partir do corpus de treinamento original e fazendo um treinamento com cada uma dessas n novas formas do corpus, seguida da combinação das saídas utilizando votação – esta classe de métodos é chamada de *arcing (adaptive resampling and combining)*.

Os métodos arcng utilizados são *bagging* (de *bootstrap aggregating*) e *boosting*. Em *bagging* são formados vários corpus de treinamento do mesmo tamanho do corpus original. Cada corpus é formado selecionando-se randomicamente exemplos do corpus de treinamento original. Com isso algumas das períodos que apareciam no corpus original podem não aparecer no novo corpus e outras podem aparecer duplicadas. E, então, é feito um treinamento com cada um dos corpus gerados e um único algoritmo de etiquetagem. Cada etiquetador gerado dá seu voto e o voto da maioria ganha. Apesar dos vários modelos serem gerados a partir do mesmo conjunto de treinamento, ao contrário do que se espera, os modelos originados usualmente não são praticamente idênticos e não classificam as novas instâncias da mesma forma, principalmente se o conjunto de treinamento é pequeno. Com isso, *bagging*, geralmente, gera um modelo significativamente melhor do que um modelo de um classificador simples (sem combinação) e nunca gera um modelo substancialmente pior (Witten & Frank, 2000). *Bagging* explora a instabilidade inerente em algoritmos de aprendizado, o que faz com que não funcione com algoritmos de classificação estáveis, que não são sensíveis a pequenas mudanças na entrada, isto porque nestes casos pequenas mudanças nos dados de treinamento não causam mudanças nos modelos. A combinação funciona justamente por explorar as diferenças entre estes modelos que fazem com que um modelo complemente o outro.

Boosting também gera vários corpus de treinamento do corpus original. A diferença é que eles não são feitos em separado como em *bagging*. Aqui cada corpus de treinamento é influenciado pelo desempenho do modelo originado com o treinamento do corpus anterior. *Boosting* tenta encorajar os novos modelos a serem especialistas nas etiquetas escolhidas erroneamente pelos modelos anteriores. No primeiro corpus todos os períodos começam com um mesmo peso, o etiquetador é então treinado com este corpus e o novo corpus será gerado diminuindo o peso das etiquetas que foram classificadas corretamente e aumentando o peso das

etiquetas que foram classificadas erradamente. Os votos de cada modelo não tem o mesmo peso, tem seu peso estabelecido por seu desempenho.

Neste trabalho, foi implementado apenas *bagging* utilizando o TreeTagger como trígrama. A escolha do TreeTagger foi devido ao fato deste algoritmo ser baseado em árvore (e portanto ser instável) e ter um custo de treinamento baixo. Não foi encontrado na literatura de combinação de classificadores um número padrão que devesse ser utilizado como número de conjuntos a serem gerados a partir do conjunto de treinamento inicial e sim apenas os valores utilizados por cada autor em seu trabalho. Quinlan (1996) gera 10 conjuntos, Bauer & Kohavi (1999) 25, Breiman (1996b) 50 e Freund & Schapire (1996) 100. Resolvemos então realizar experimentos com três destes quatro valores — 10, 25 e 50 — verificando assim qual é o melhor para a nossa tarefa, a etiquetagem morfossintática. A precisão obtida em cada um dos experimentos é mostrada na Tabela 32.

Tabela 32 - Precisão da combinação usando *bagging*

Número de corpus gerados a partir do corpus de treinamento original (C2)	Precisão
10	89,32%
25	89,40%
50	89,65%

5.2.3 Método em cascata aplicado ao TBL

Não encontramos sugestões detalhadas na literatura para o modelo em cascata. Fala-se apenas que os dados classificados por um classificador X servirão como entrada para um classificador Y. Mas não há sugestões, por exemplo, de quantos classificadores encadear.

Uma forma possível no caso da etiquetagem morfossintática seria, por exemplo, utilizar a etiqueta dada como resposta pelo etiquetador inicial para casos em que o etiquetador seguinte não consiga etiquetar aquela palavra. Pode-se pensar também em um etiquetador que admita ter como entrada um texto já com uma etiquetagem inicial e que dado este texto tente melhorar a etiquetagem.

A opção utilizada neste trabalho foi a última acima. O etiquetador TBL admite também a opção de que não seja utilizado seu etiquetador inicial, podendo fornecer a ele um texto já etiquetado para ser passado pelo seu etiquetador final, o etiquetador contextual (que foi explicado no Capítulo 3, na Seção 3.3.1.2). Foram feitos três experimentos:

- 1) O corpus de teste (C31) foi etiquetado pelo etiquetador TreeTagger como unígrafo (treinado com C3) e depois fornecido como entrada para o etiquetador TBL
- 2) O corpus de teste (C31) foi etiquetado pelo etiquetador TreeTagger como trígrafo (treinado com C3) e depois fornecido como entrada para o etiquetador TBL
- 3) O corpus de teste (C31) foi etiquetado pelo etiquetador MXPOST (treinado com C3) e depois fornecido como entrada para o etiquetador TBL

Nos experimentos 1 e 2 a precisão foi maior do que havia sido obtida pelo TreeTagger, mas não superou a precisão do etiquetador TBL. Apenas no experimento 3 a precisão superou a precisão obtida pelo etiquetador TBL e a obtida pelo etiquetador inicial – MXPOST. A Tabela 33 mostra os resultados obtidos com os experimentos.

Tabela 33 - Resultados obtidos nos modelos em cascata

Etiquetadores	Precisão	Precisão do modelo em cascata
TBL	88,76%	-
TreeTagger como unígrafo	80,01%	80,79%
TreeTagger como trígrafo	88,47%	88,70%
MXPOST	89,66%	89,68%

6 DISCUSSÃO DOS RESULTADOS

Primeiro obtenha os fatos; depois pode torcê-los tanto quanto quiser. - Mark Twain

Como mostrado no Capítulo 4 a melhor precisão geral individual é de 89,66% (melhor etiquetador no corpus de teste), que é um valor insatisfatório, não só por não alcançar os valores obtidos pelos mesmos etiquetadores quando treinados para o inglês, mas principalmente pelo que uma taxa de erro de aproximadamente 10% significa. Dizer que um etiquetador tem 90% de precisão geral é o mesmo que dizer que ele comete em média três erros por período, considerando que um período tenha cerca de 30 palavras. Dado que a etiquetagem morfossintática é uma tarefa básica para a maioria dos aplicativos de PLN, começar com esta taxa de erro em cada período pode ser uma forte desvantagem, especialmente se considerarmos que alguns destes erros podem crescer mais que linearmente. Alguns fatores poderiam ser os responsáveis por esta baixa precisão:

- qualidade da etiquetagem manual do corpus de treinamento;
- tamanho do corpus de treinamento;
- o conjunto de etiquetas;
- os tipos de texto utilizados;
- as abordagens de etiquetagem.

Fora os experimentos citados até aqui, vários outros foram feitos a cada nova versão corrigida do corpus manualmente etiquetado. Após o treinamento e teste dos etiquetadores, uma

lista com as situações em que todos eles davam a mesma resposta e estavam errados era gerada junto com a resposta da etiquetagem manual para que pudessemos assim encontrar possíveis erros de etiquetagem manual. Assim, ao longo dos experimentos foram revisadas as palavras "que", "se", "como", "outro", "algum", "todo", "qualquer", "cada", "vários", "muitos" e "poucos", as conjunções, locuções e verbos. As correções dos problemas de padronização na etiquetagem manual ocasionaram um aumento de 0,93% na precisão geral da penúltima para a última versão do corpus etiquetado manualmente.

O tamanho do corpus de treinamento foi um dos fatores de maior influência para a baixa precisão encontrada. Como mostrado na Seção 4.2.1.3, incrementando o corpus de treinamento com 10.500 palavras a precisão saltou de 89,66% para 90,25%, ou seja, houve um aumento de 0,59%. O que faz do tamanho do corpus um fator tão importante, é o fato que quanto maior o corpus menor o número de palavras e de contextos desconhecidos. Ainda na Seção 4.2.1.3 foi mostrada a influência das palavras desconhecidas, e viu-se que quando não existiam palavras desconhecidas a precisão geral foi de 93,65%, um aumento de 3,99%. Esta relação entre tamanho do corpus versus número de palavras desconhecidas pode ser vista na Tabela 8 da Seção 4.2. O que mostra que quando se muda a divisão do corpus de 90-10 para 80-10-10 o número de palavras desconhecidas aumenta em cerca de 1%. É importante ressaltar também que nos experimentos para o inglês feitos em (van Haltaren et al., 1998), a porcentagem de palavras novas é de 2,5% e a porcentagem de palavras com etiquetas novas é de 0,2%. Já em nossos experimentos, a porcentagem de palavras desconhecidas no corpus de teste é de 15,21% e a porcentagem de palavras com etiquetas novas é 0,2.

O Nilc Tagset com certeza é outro causador da baixa precisão. Na Seção 4.2.2 quando treinamos o etiquetador MXPOST com o mesmo corpus de treinamento, porém sem considerar a transitividade, a precisão aumentou para 92,51% – um aumento de 2,85% na precisão geral. Nas Seções 4.2.1.1, 4.2.1.2 e 4.2.1.3, fica claro também que as etiquetas referentes às locuções, palavra denotativa, ênclices e mesóclises são as mais problemáticas, mas não são as grandes vilãs, primeiro por causarem uma pequena parte dos erros (aparecem muito pouco nos textos), segundo por serem casos fáceis de resolver, não justificando uma alteração no conjunto de etiquetas. O problema com as locuções pode ser resolvido com o pós-processamento através de uma lista fechada. A classe das palavras denotativas exige bastante cuidado devido à ambigüidade com adjetivos, advérbios, locuções, preposições e conjunções. Porém, segundo Celso Cunha (1979) as

palavras denotativas não devem ser incluídas entre os advérbios, posto que não modificam o verbo, nem o adjetivo, nem outro advérbio. Resolve-se assim a classe de ambigüidade formada por palavra denotativa-advérbio. Há a exceção das palavras “cá” e “lá”, que exercem a função de advérbio e que têm uso raro como palavras denotativas no português do Brasil, por exemplo: “Tenho cá minha opinião”. No entanto, mesmo em nosso corpus manualmente etiquetado, esta não foi a metodologia utilizada para etiquetar as palavras denotativas. Já os erros relativos às mesóclises e ênclises estão relacionados ao número reduzido de vezes em que aparecem em nosso corpus, o que impossibilitou que os etiquetadores aprendessem regras simples²⁵ para tratar parte destes casos como, por exemplo, as mostradas na Figura 37.

Se a palavra terminar em: -o, - a, - os, - as, - no, - na, - nos, - nas, ela deve ser etiquetada como VTD +PPOA
Se a palavra termina em: -lo-x, -la-x, -los-x, las-x , ela deve ser etiquetada como VTD!PPOA x = {ei, á, ás, emos, eis, ão, ia, ias, íamos, ieis, iam}
Se a palavra termina em -se: ela deve ser etiquetada como VTD+PPOA quando a palavra anterior for N ou NP ela deve ser etiquetada como VTD+PAPASS quando iniciar a frase ela deve ser etiquetada como VINT+PREAL quando a próxima palavra/símbolo for , ou . ou adv ela deve ser etiquetada como VTI+PPOA quando a próxima palavra for a palavra a ela deve ser etiquetada como VTI+PPOA quando a próxima palavra for N ou for a palavra de ela deve ser etiquetada como VTI+PPOA quando as próximas palavras forem: ART N ou ART ADJ N
Se a palavra terminar em: -lhe, -lhes , ela deve ser etiquetada como VTI+PPOA
Se a palavra termina em: -lhe-x, -lhes-x, ela deve ser etiquetada como VTI!PPOA x = {ei, á, ás, emos, eis, ão, ia, ias, íamos, ieis, iam}
Se a palavra termina em -nos, -vos, -me, -te, deve-se procurar no léxico pela palavra sem estas terminações, em caso da palavra encontrada ser VTD deve-se etiquetar a palavra original como VTD+PPOA e em caso de ser VTI deve-se etiquetar a palavra original como VTI+PPOA.

Figura 37 - Regras para pós-preprocessamento de ênclises e mesóclises

Quais textos utilizar foi outra escolha problemática que certamente influenciou na precisão geral obtida nos experimentos. Os experimentos para inglês reportados na literatura geralmente fazem uso de corpus formados por textos jornalísticos. Já nosso corpus contém aproximadamente 30,54% de textos literários, que, como mostrado na Tabela 7 da Seção 4.2, é o tipo de texto com proporcionalmente a maior taxa de ambigüidade.

Como os etiquetadores utilizados neste trabalho são reconhecidos pela literatura como estado-da-arte, o problema é mais uma vez o tamanho do corpus, já que estes etiquetadores precisam de corpus grandes (entre 1.000.000 e 2.000.000 de palavras) para obterem bons

²⁵ Estas regras fazem parte das regras elaboradas por uma lingüista para o PoSiTagger.

resultados. Uma solução seria utilizar uma abordagem que trabalhasse bem com corpus pequenos, como é o caso dos etiquetadores neurais. Outra seria aumentar progressivamente o tamanho do corpus – treinando, etiquetando automaticamente, corrigindo manualmente e retreinando, até que se obtivesse um corpus etiquetado com 1 milhão de palavras. Ou ainda combinar os etiquetadores como foi feito no Capítulo 6.

O etiquetador simbólico PoSiTagger também não atingiu a precisão imaginada, já que para outras línguas este é o tipo de etiquetador que atinge a maior precisão geral, pelo menos quando restritos a um domínio. Atribuímos esta baixa precisão ao tempo destinado para elaboração das regras — 20 dias —, e ao fato destas ainda não terem sido reavaliadas e refinadas através de testes.

A combinação de etiquetadores é uma solução bem interessante para se obter etiquetadores estáveis. No entanto, apenas seis dos métodos de combinação mostrados na Seção 5.2.1 ultrapassaram a precisão geral do melhor dos etiquetadores com um acréscimo máximo de 1,25% sobre a precisão geral do melhor deles, enquanto que para o inglês todos os métodos de combinação superaram a precisão geral do melhor etiquetador. Em todos os experimentos de combinação em que se utilizou bagging os resultados foram melhores que o do etiquetador individual (TreeTagger como trígrama), com um acréscimo de até 1,18% na precisão, no entanto, não houve melhora com relação ao melhor dos etiquetadores. E, com o modelo em cascata o acréscimo na precisão foi mínimo – 0,02%.

7 CONCLUSÕES E TRABALHOS FUTUROS

Experiência é o nome que cada qual dá a seus próprios erros. — Oscar Wilde

Com todos os experimentos realizados fica evidente a importância de um bom corpus de treinamento – um corpus grande e com uma taxa de erro de etiquetagem manual mínima. Entretanto, a construção de um corpus etiquetado é uma tarefa que exige tempo e cuidados. Neste trabalho, a etiquetagem manual foi feita em paralelo com a construção dos etiquetadores. Em Alves (1999) há uma analogia que descreve muito bem o processo de construir um etiquetador enquanto se constrói um corpus etiquetado, comparando este processo a "tentar navegar em um navio enquanto é construído". Apesar disso, este trabalho gerou várias contribuições (Seção 7.1). A Seção 7.2 traz algumas sugestões de trabalhos futuros.

7.1 Contribuições

Apesar da baixa precisão geral, este trabalho contribuiu para a etiquetagem automática comprovando que:

- É possível utilizar um conjunto de etiquetas que contenha também outras características além das morfossintáticas, como é o caso do Nilc tagset que trata também a transitividade de verbos. Apesar da literatura dizer que as etiquetas podem ser de outra natureza que não a

morfossintática, não encontramos na literatura conjuntos de etiquetas que considerassem características morfossintáticas e de outro tipo ao mesmo tempo.

- A combinação de etiquetadores melhora a precisão mesmo com corpus pequenos.
- O uso de diferentes tipos de texto no treinamento e etiquetagem influencia consideravelmente na precisão geral. Não encontramos trabalhos que tivessem feito este teste mesmo que fazendo parte das recomendações do EAGLES (1996b). Encontramos apenas um trabalho que identificou a influência causada na precisão por textos escritos em épocas diferentes (Alves, 1999).
- É importante o uso de técnicas estatísticas se para estimar a taxa de erro verdadeira de etiquetadores. Notou-se que a precisão já variava entre os arquivos de teste e calibração. E o experimento utilizando *bootstrap* mostrado na Seção 4.2.1.1 mostrou a diferença entre a precisão encontrada no teste e a estimada como sendo a real (88,47%-88,95%, respectivamente).

7.2 Trabalhos futuros

São vários os trabalhos que podem surgir deste projeto de mestrado. Listamos alguns abaixo que são caracterizados como extensões e como superação das limitações encontradas.

- Treinar o etiquetador neural elático (Ma et al., 1999) com 90% do corpus manualmente etiquetado. Foi implementada uma ferramenta para que o corpus fosse colocado no formato exigido por este etiquetador (Apêndice D).
- Treinar o etiquetador TnT (Brants, 2000) com 90% do corpus manualmente etiquetado.
- Treinar os etiquetadores TreeTagger, TBL e MXPOST com 90% do corpus.
- Analisar as regras para palavras iniciais e as regras para palavras contextuais do etiquetador TBL, na tentativa de melhorar sua precisão.
- Refinar as regras do etiquetador PoSiTagger — testando, avaliando e reescrevendo as regras problemáticas.
- Verificar se o PoSiTagger trata as classes de ambigüidade das palavras que forem as maiores responsáveis pela taxa de ambigüidade.
- Utilizar o etiquetador PoSiTagger nos experimentos de combinação.

- Refazer os experimentos com combinação do Capítulo 6 utilizando mais etiquetadores (treinados com 90% do corpus manualmente etiquetado) e no caso dos algoritmos da Seção 5.2.1 utilizar o corpus de treinamento como corpus de calibração.
- Utilizar boosting para combinação.
- Treinar o etiquetador TBL (Brill, 1994a) não-supervisionado com 90% do corpus manualmente etiquetado.
- Fazer experimentos com etiquetadores totalmente não supervisionados.
- Aumentar o corpus de treinamento de 100.000 em 100.000 até alcançar um corpus de 1 milhão de palavras – etiquetando textos com 100.000 palavras com o melhor dos algoritmos de combinação, corrigindo estes textos e retreinando todos os etiquetadores com o novo corpus.
- Implementar outras das técnicas estatísticas para estimar a taxa de erro verdadeira mostradas na Seção.
- Etiquetar períodos escritos de forma incorreta para verificar o quanto estes erros influenciam no trabalho do etiquetador.
- Como os textos que são etiquetados dificilmente estão livres de erros, verificar se o aprendizado na presença de erros como é feito em Aprendizado de Máquina (Kearns, 1990) melhorará a precisão geral.
- Verificar se alguns problemas do ReGra são resolvidos quando são utilizados textos etiquetados.

BIBLIOGRAFIA E REFERÊNCIAS

- (Abney et al., 1999) Abney, S.; Shapire, R. E.; Singer, Y. Boosting Applied to Tagging and PP Attachment. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999. p. 38-41.
- (Abney, 1997) Abney, S. *Part-of-Speech Tagging and Partial Parsing. Corpus-Based Methods in Language and Speech Processing*. Editado por Steve Young e Gerrit Blothoof. Kluwer Academic Publishers, 1997. p. 118-136.
- (Academia Brasileira de Letras, 1998) Academia Brasileira de Letras. *Vocabulário Ortográfico da Língua Portuguesa*. 2^a edição. Imprensa Nacional, 1998.
- (Agirre et al., 1998) Agirre, E.; Gojenola, K.; Sarasola, K.; Voutilainen, A. Towards a Single Proposal in Spelling Correction. In *COLING-ACL '98*, p. 22-28.
- (Ali & Pazzani, 1996) Ali, K. M.; Pazzani, M. J. Error Reduction through Learning Multiple Descriptions. *Machine Learning*, 24(3):173-202.
- (Alpaydin, 1998) Alpaydin, E. Techniques for Combining Multiple Learners. In *Proceedings of Engineering of Intelligent Systems*, p. 6-12.
- (Alves & Finger, 1999) Alves, Carlos D. e Finger, Marcelo. Etiquetagem do Português Clássico Baseada em Córpora. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Évora, 1999. p. 17-31.
- (Alves, 1999) Alves, Carlos Daniel Chacur. *Etiquetagem do Português Clássico Baseada em Corpus*. São Paulo, 1999. Dissertação (Mestrado) – IME - Universidade de São Paulo.
- (André, 1990) André, Hildebrando A. de. *Gramática Ilustrada*. Editora Moderna. 4^a edição. 1990.
- (Armstrong et al., 1995) Armstrong, S.; Russel, G.; Petitpierre, D.; Robert, G. An Open Architecture for Multilingual Text Processing. In: *SIGDAT-95 (EACL-95 Workshop)*, Dublin, 1995. p. 30-34.
- (Augusto et al, 1998) Augusto, M.; Britto, H.; Scher, A. P. *Morphological tagging for different periods of*

- statistical and a constraint-based method.* 1995. (Disponível em <http://xxx.lanl.gov/cmp-lg/9503003>).
- (Charniak & Wilks, 1976) Charniak, E.; Wilks, Y. *Fundamental Studies in Computer Science 4: Computational Semantics*. North-Holland Publishing Company, 1976. p. 89-100, 155-184.
- (Charniak et al., 1993) Charniak, E.; Hendrickson, C.; Jacobson, N.; Perkowitz, M. Equations for part-of-speech tagging. In: *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993. p. 784-789.
- (Charniak et al., 1994) Charniak, E.; Carroll, G.; Adcock, J.; Cassandra, A.; Gotoh, Y.; Katz, J.; Littman, M.; MCCANN, J. *Taggers for Parsers. Technical Report CS-94-06*, Brown University, 1994.
- (Charniak, 1995) Charniak, E. *Statistical language learning.* 1995. (Disponível em <http://xxx.lanl.gov/cmp-lg/9506019>)
- (Chen et al., 1999) Chen, J.; Bangalore, S.; Vijay-Shanker, K. New Models for Improving Supertag Disambiguation. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, p. 188-195.
- (Chomsky, 1995) Chomsky, N. *The minimalist program*. Cambridge, Mass., The MIT Press, 1995.
- (Church & Gale, 1991) Church, K. W.; Gale, W.A. A comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams. *Computer Speech and Language*, n.5, p. 19-54, 1991.
- (Church & Gale, 1991) Church, k.; Gale, W. Concordances for parallel text. In *Proceedings, Seventh Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*. 1991.
- (Church, 1988) Church, K. A Stochastic Parts Program and Noun Phrase Parser for Unrestrict Text. In: *Proceedings ANLP 88*, Austin, TX, 1988.
- (Cole, 1996) Cole, R. (Ed.) *Survey of the State of the Art in Human Language Technology*. Center for Spoken Language Understanding, Oregon Graduate Institute, 1996. (Disponível em <http://www.cse.ogi.edu/cslu/HLTsurvey>)
- (Cunha & Cintra, 1985) Cunha, C. e Cintra, L. *Nova Gramática do Português Contemporâneo*. Rio de Janeiro: Nova Fronteira, 1985.
- (Cunha, 1979) Cunha, C. Ferreira da. *Gramática da Língua Portuguesa*. Rio de Janeiro, Fename, 1979. 5^a edição.
- (Cutting et al., 1992) Cutting, D., Kupiec, J., Pedersen, J.,; Sibun, P. A Practical Part-of-Speech Tagger. In: *Third Conference on Applied Natural Language Processing (ANLP-92)*. Trento, 1992. p. 133-140.
- (d'Andrade, 1993) d'Andrade, Ernesto. *Dicionário Inverso do português*. Edições Cosmos. 1993. 1^aedição.
- (Daelemans et al, 1996) Daelemans, W.; Zavrel, J.; Berck, P.; Gillis, S. MTB: A Memory-Based Part of Speech Tagger-Generator. In: *Proceedings of Fourth WLC*. Copenhagen, 1996. P. 14-27.
- (Daelemans et al, 1997) Daelemans, W.; Bosch, ^a Van den; Weijters, A. IGTTree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407-423. 1997.
- (Dermatas & Kokkinakis, 1995) Dermatas, E.; Kokkinakis, G. Automatic Stochastic Tagging of Natural Language Texts. *Computational Linguistics*, 21(2):137-164. 1995.

- (Brill, 1992) Brill, E. A Simple Rule-Based Part of Speech Tagger. *In: Proceedings of the Third Conference on Applied Natural Language Processing (ACL)*. Trento, Italy, 1992. P.152-155.
- (Brill, 1993a) Brill, E. Automatic Grammar Induction and Parsing Free-Text: A Transformation-Based Approach. *In: Proceedings of Association For Computational Linguistics Annual Meeting (ACL-93)* 31., Columbus.OH, 1993. (Disponível em <http://www.cs.jhu.edu/~brill/papers.html>)
- (Brill, 1993b) Brill, E. *A Corpus-Based Approach to Language Learning*. Philadelphia, 1993. Dissertation (PhD) - University of Pennsylvania.
- (Brill, 1994a) Brill, E. Some Advances in Transformation-Based Part-of-Speech Tagging. *In: Proceedings of National Conference On Artificial Intelligence (AAAI-94)*, 12., Seattle, 1994. (Disponível em <http://www.cs.jhu.edu/~brill/papers.html>)
- (Brill, 1994ab) Brill, E. A Rule-Based Approach to Prepositional Phrase Attachment Disambiguation. *In: Proceedings of COLING 1994*. (Disponível em <http://www.cs.jhu.edu/~brill/papers.html>)
- (Brill, 1995) Brill, E. Transformation-based error-driven learning of natural language: A case study in part of speech tagging. *Computational Linguistics*, v. 21, n.4, p. 543-565, 1995. (Disponível em <http://www.cs.jhu.edu/~brill/papers.html>)
- (Brill, 1996a) Brill, E. Efficient Transformation-Based Parsing. *In: Proceedings of ACL 1996*. (Disponível em <http://www.cs.jhu.edu/~brill/papers.html>)
- (Brill, 1996b) Brill, E. Learning to Parse With Transformations. *In: Recent Advances in Parsing Technology*, Kluwer Academic Publishers. 1996. (Disponível em <http://www.cs.jhu.edu/~brill/papers.html>)
- (Brill, 1997a) Brill, E. Automatic Rule Acquisition for Spelling Correction. *In: Proceedings of ICML 97*, 1997. (Disponível em <http://www.cs.jhu.edu/~brill/papers.html>)
- (Brill, 1997b) Brill, E. Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging. *In: Natural Language Processing Using Very Large Corpora*. Kluwer Academic Press, 1997. (Disponível em <http://www.cs.jhu.edu/~brill/papers.html>)
- (Briscoe et al., 1994) Briscoe, Ted; GREFENSTETTE, Greg; PADRÓ, Lluís; SERAIL, I. Hybrid techniques for training HMM part-of-speech taggers. *Acuilex II working paper 45*, 1994.
- (Britto et al.) Britto, H. et al. *Morphological annotation system for automated tagging of electronic textual corpora: from English to romance languages*. (Disponível em <http://www.ime.usp.br/~tycho/papers/index.html>).
- (Calzolari & Monachini, 1994) Calzolari, N.; Monachini, M. MULTEXT: *Lexical specifications: application to Italian*. Pisa, ILC. (Disponível em <http://www.lpl.univ-aix.fr/projects/multext/LEX/LEX.LangSpec.it.html>).
- (Chan et al., 1999) Chan, P. K.; Stolfo, S. J.; Wolpert, D. Guest Editors'Introduction. Special Issue on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms. *Machine Learning*, 36(1-2):5-7.
- (Chanod & Tapanainen, 1995a) Chanod, J.p.; Tapanainen, P. *Creating a tagset, lexicon and guesser for a French tagger*. 1995. (Disponível em <http://xxx.lanl.gov/cmp-lg/9503004 v2>).
- (Chanod & Tapanainen, 1995b) Chanod, J.p.; Tapanainen, P. *Tagging French – comparing a*

- (Harris, 1982) Harriz, Z. *A grammar of English on mathematical principles*. New York: Wiley, 1982.
- (Hays, 1966) Hays, D. G. *Readings in automatic language processing*. American Elsevier Publishing Company Inc, 1966. p. 73-82.
- (Hearst, 1991) Hearst, M. Toward noun homonym disambiguation using local context in large text corpora. In *Proceedings, Seventh Annual Conference of the UW Centre for the New OED and Text Research*. University of Waterloo, Waterloo, Ontario, p. 1-22. 1991.
- (Ho et al., 1994) Ho, T. K.; Hull, J. J.; Srihari, S. N. Decision Combination in Multiple Classifier Systems, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, No. 1, p. 66-75.
- (Hutchinson, 1994) Hutchinson, Alan. Algorithmic Learning. Clarendon Press, 1994.
- (Ibaixe, 1994) Ibaixe, Ana Maria. Tópicos de Linguagem – Gramática. As palavras Que, Se e Como – Teoria e Prática. Atual Editora. 6^aedição. 1994.
- (Jelinek & Mercer, 1980) Jelinek, F; Mercer, F. L. Interpolated estimation of Markov source parameters from sparse data. In Proceedings of the Workshop on Pattern Recognition in Practice, 381-397. 1980.
- (Jelinek et al., 1991) Jelinek, F.; Mercer, R.; Roukos, S. Principles of lexical language modeling for speech recognition. In *Advances in Speech Signal Processing*. Marcel Dekker, p. 651-700. 1991.
- (Jelinek, 1985a) JELINEK, F. Robust part of speech tagging using a hidden markov model. 1985. *Technical Report*, IBM, T.J. Watson Research Center.
- (Jelinek, 19985b) Jelinek, F. Self-organized language modeling for speech recognition. IBM Report. Reprinted in W&L, 450-506.
- (Karlsson, 1990) Karlsson, F. Constraint Grammar as a Framework for Parsing Running Text. In *COLING-90*. Helsinki, 1990. p. 168-173.
- (Karttunen, 1994) Karttunen, L. Constructing Lexical Transducers. In: *Proceedings of Coling-94. The fifteenth International Conference on Computational Linguistics*. Vol I, p. 406-411. Kyoto, 1994.
- (Kearns, 1990) Kearns, Michael J. The Computational Complexity of Machine Learning. MIT Press, 1990.
- (Kearns, 1994) Kearns, Michael J. An Introduction to Computational Learning Theory. MIT press, 1994.
- (Kelly, 1992) Kelly, M. Using sound to solve syntactic problems: the role of phonology in grammatical category assignments. *Psychological Review*, v 99, n 2, p. 349-364. 1192.
- (Kempe, 1993) Kernpe, André. Stochastic Tagger and an Analysis of Tagging Errors. Stuttgart: University of Stuttgart, 1994. Technical Report.
- (Kempe, 1994) Kempe, André. Probabilistic Tagging with Feature Structures. In: *Proceedings of the International Conference on Computational Linguistics - COLING*. 1994
- (Kim & Norgard, 1998) Kim, Youngin; Norgard, Barbara. Adding Natural Language Processing Techniques to the Entry Vocabulary Module Building Process. 1998.
- (Disponível em <http://metaphor.sims.berkeley.edu/papers/nlptech.html>
- (Klavans & Tzoukermann, 1990) Klavans, J.; Tzoukermann, E. Linking bilingual corpora and machine readable dictionaries with the BICORD system. In *Proceedings, Sixth Annual Conference of the UW*

- (Deroault & Merialdo, 1984) Deroault, A. M.; Merialdo, B. Language modelling at the syntactic level. In: *Proceedings 7th International Conference on Pattern Recognition*, 1984.
- (Derose, 1988) Derose, S. Grammatical category disambiguation by statistical optimisation. *Computational Linguistics*, Cambridge, v. 14, n.1, p. 31-39, 1988.
- (Dietterich, 1997) Dietterich, T. G. Machine Learning Research: Four Current Directions. *AI Magazine*, 18(4):97-136.
- (EAGLES, 1996a) EAGLES - Expert Advisory Group on Language Engineering Standards. *Study of the relation between Tagsets and Taggers*. 1996. (Disponível em <http://www.ilc.pi.cnr.it/EAGLES96/browse.html>).
- (EAGLES, 1996b) EAGLES - Expert Advisory Group on Language Engineering Standards. *Recomendations for the Morphosyntactic Annotation of Corpora*. 1996. (Disponível em <http://www.ilc.pi.cnr.it/EAGLES96/browse.html>).
- (Faraco & Moura, 1994) Faraco, Carlos E. e Moura, Francisco M. de M. *Gramática- Faraco & Moura*. Editora Ática. 1994. 13^a edição.
- (Feldweg, 1995) Feldweg, H. *Implementation and evaluation of a German HMM for POS disambiguation*. 1995. (Disponível em <http://xxx.lanl.gov/cmp-lg/9502038>).
- (Fernandes, 1995) Fernandes, Francisco. *Dicionário de Verbos e Regimes*. 40^a edição. Editora Globo. 1995.
- (Finger, 1998) Finger, M. *Tagging a Morphologically Rich Language: The Construction of the Tycho Brahe Parsed Corpus of Historical Portuguese*. IME, Universidade de São Paulo. (Disponível em <http://www.ime.usp.br/~tycho>).
- (Francis & Kucera, 1979) Francis, W. N.; Kucera, H. *Brown Corpus Manual*. Providence, Rhode Island, Department of Linguistics, Brown University, 1979. (Disponível em <http://www.hit.uib.no/icame/brown/bcm.html>)
- (Freund & Schapire, 1996) Freund, Y. & Schapire, R. E. Experiments with a new boosting algorithm. In L. Saitta (Ed.), *Machine Learning*. Proceedings of the Thirteenth National Conference (p. 148-156). Morgan Kaufmann. 1996.
- (Galves & Britto, 1999) Galves, Charlotte. e Britto, Helena. A construção do Corpus Anotado do Português Histórico Tycho Brahe: o sistema de anotação. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Évora, 1999. p. 81-92.
- (Garside) Garside, R. *Using CLAWS to annotate the British National Corpus*. Department of Computing, University of Lancaster. (Disponível em http://info.ox.ac.uk/bnc/what/garside_allc.html)
- (Golding & Roth, 1999) Golding, A. R.; Roth, D. A Winnow-Based Approach to Context-Sensitive Spelling Correction. *Machine Learning*, 34(1-3):107-130.
- (Greene & Rubin, 1971) Greene, B. B.; Rubin, G. M. *Automated Grammatical Tagging of English*. Department of Linguistics, Brown University.
- (Hagège & Bès, 1998) Hagège, C. e Bès, G. G. Da observação de propriedades lingüísticas a sua formalização numa gramática do processamento da língua. In: *Anais do III Encontro para o Processamento Computacional de Português Escrito e Falado (PROPOR'98)*. Porto Alegre, 1998. p. 23-31.

(Disponível em <http://www.ldc.upenn.edu/doc/treebank2/d93.html~>)

(Marques & Lopes, 1996) Marques, N. C.; Lopes, G. Pereira. A Neural Network Approach to Part-of-Speech Tagging. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Curitiba, 1996. p. 1-9.

(Marques, 1995) Marques, N. C. YATH - Yet another HMM Tagger - Modelos de Markov Escondidos Aplicados à Classificação de Largos Corpora de Textos. Technical Report, Universidade Nova de Lisboa, 1995.

(Màrquez et al., 2000) Márquez, Lluís; Padró, Lluís; Rodríguez, Horacio. A Machine Learning Approach to POS Tagging. *Machine Learning*, 39, 59-91, 2000. Kluwer Academic Publishers.

(Mendes & Eizirik, 1989) Mendes, S. Bandeira Teixeira e Eizirik, L.m. Ripoll. Um modelo neuronal probabilístico para solução da ambigüidade do português. *Revista brasileira de computação*, Rio de Janeiro, v. 5, n.1, p. 45-51 , Jul./Set. 1989.

(Merialdo, 1994) Merialdo, B. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, Cambridge, v.20, n.2, p.155-177, 1994.

(Mikheev, 1996) Mikheev, A. *Unsupervised Learning of Word-Category Guessing Rules*. 1996. (Disponível em <http://xxx.lanl.gov/cmp-lg/9604022>)

(Moses, 1986) Moses, Lincoln E. Think and Explain with Statistics. Addison-Wesley Publishing Company, 1986.

(Nakamura & Shicano, 1989) Nakamura, M.; Shicano, K. A study of English word prediction based on neural network. In: *Proceedings of ICASSP*. Glasgow, 1989. p.731-734.

(Nakamura et al.) Nakamura, M. et al. Neural Network Approach to word Category Prediction for English Texts. In: *Proceedings of COLING-90*, 1990, Helsinki University. p. 213-218.

(NILC, 1999) NILC _ Núcleo Interinstitucional de Lingüística Computacional. O projeto ReGra. In: *Introdução ao processamento das línguas naturais*. ICMC-USP, São Carlos, Março 1999. p. 77-80.

(NLP) NLP – Group of the School of Computer Studies at Leeds University. *Automatic Mapping Among Lexico-Grammatical Annotation Models (AMALGAM)*. (Disponível em (Disponível em <http://www.scs.leeds.ac.uk/amalgam/amalgam/amalghome.htm>)

(Nunes et al., 1996a) Nunes, Maria das Graças V.; Vieira, F. M. C.; Zavaglia, C.; Sossolote, C. R. C. e Hernandez, J. A construção de um léxico para o Português do Brasil: Lições aprendidas e perspectivas. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Curitiba, 1996. p. 61-70.

(Nunes et al., 1996b) Nunes, Maria das Graças V.; Ghiraldello, C. M.; Montilha, G.; Turine, M. A. S.; Oliveira, M. C. F.; Hasegawa, R.; Martins, R. T. e Oliveira Jr, O. N. Desenvolvimento de um Sistema de Revisão Gramatical Automática para o Português do Brasil. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Curitiba, 1996. p. 71-80.

(Nunes et al., 1996c) Nunes, Maria das Graças V. *A Construção de um léxico para suporte à revisão automática do Português do Brasil*. Relatório Técnico do ICMC-USP, nro 42, 1996, 37p.

(Oflazer & Kuruöz, 1994) Oflazer, K.; Kuruöz, I. Tagging and Morphological Disambiguation of Turkish Text. In: *Fourth Conference on Applied Natural Language Processing (ANLP-94)*. Stuttgart, 1994. p.144-149.

(Oksefjell & Santos, 1998) Oksefjell, S. e Santos, D. Breve Panorâmica dos Recursos de Português

- Centre for the New Oxford English Dictionary and Text Research, p. 19-30. 1990.
- (Klein & Simmons, 1963) Klein, S.; Simmons, R. A Computational Approach to Grammatical Coding of English Words. *JACM*, 10:334-337.
- (Koskenniemi et al., 1992) Koskenniemi, K.; Tapanainen, P.; Voutilainen, A. Compiling and using finite-state syntactic rules. In: *Proceedings of the fifteenth International Conference on Computational Linguistics. COLING-92*. Vol. I, p. 156-162, Nantes, France.
- (Koskenniemi, 1990) Koskenniemi, K. Finite-state parsing and disambiguation. Proceedings of the fourteenth International Conference on Computational Linguistics. *COLING-90*. Helsinki, Finland, 1990.
- (Kupiec, 1992) Kupiec, J. Robust part-of-speech tagging using a hidden markov model. *Computer Speech and Language*, 6:225-242.
- (Kury, 1993) Kury, Adriano da Gama. Novas Lições de Análise Sintática. Editora Ática. 6^a edição. 1993.
- (Leech et al., 1994) Leech, G.; Garside, R.; Bryant, M. Claws4: The tagging of the British National Corpus. In: *Proceedings of the 15th International Conference on Computational Linguistics, (COLING94)*, 1994. p. 622-628.
- (Léon & Serrano, 1995) Léon, F. S.; Serrano, A. F. N. Development of a Spanish Version of the Xerox Tagger. (Disponível em (<http://xxx.lanl.gov/cmp-lg/9505035>)).
- (Liberman & Church, 1991) Liberman, M.; Church, K. Text analysis and word pronunciation in text-to-speech synthesis. In *Advances in Speech Signal Processing*, edited by S. Furui and M. Mohan. Marcel Dekker, p. 791-832.
- (Lima, 1992) Lima, Rocha. Gramática Normativa da Língua Portuguesa. 1992. José Olympio Editora. 31^a edição.
- (Lu et al., 2000) Lu, Bao-Liang; Ma, Q.; Ichikawa, M.; Isahara, H. Massively Parallel Part of Speech Tagging Using Min-Max Modular Neural Networks.
- (Luft, 1993) Luft, Celso Pedro. Dicionário Prático de Regência Verbal. Editora Ática. 1993.
- (Ma & Isahara, 1998) Ma, Q.; Isahara, H. A multi-neuro tagger using variable lengths of contexts. In *Proceedings of COLING-ACL'98*, Montreal, p. 802-806, 1998.
- (Ma et al., 1998) Ma, Q.; Sun, M.; Isahara, H. A multi-neuro tagger applied in Chinese texts. In *Proceedings of 1998 Int. Conf. Chinese Info. Processing*, Beijing, Nov. 18-20, p. 200-207, 1998.
- (Ma et al., 1999a) Ma, Q.; Uchimoto, K.; Murata, M.; Isahara, H. Elastic Neural Networks for part of speech tagging. In: *Proceedings of IJCNN'99*, Washington DC., July, 1999.
- (Ma et al., 1999b) Ma, Q.; Lu, B. L.; Isahara, H. Part of speech tagging with min-max neural networks. In *Proceedings of IEEE SMC'99*, Tokyo, Oct. 12-15, vol 5, p. 356-360, 1999.
- (Magerman, 1995a) Magerman, D. *Statistical Language Learning - Review*. 1995. (Disponível em <http://xxx.lanl.gov/cmp-lg/>)
- (Magerman, 1995b) Magerman, D. *Statistical Decision-Tree Models for Parsing*. 1995. (Disponível em <http://xxx.lanl.gov/cmp-lg/9504030>)
- (Marcus et al., 1993) Marcus, Michell P.; Santorini, Beatrice; Marcinkiewicz, Mary Ann. Building a large annotated corpus of English: the Penn Treebank –. 1993. *Computational Linguistics*, 19(2).

- Resolução da Ambigüidade Léxica no Processamento da Linguagem Natural. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Curitiba, 1996. P. 149-158.
- (Tapanainen, 1992) Tapanainen, P. "Äärellisiin automaatteihin perustava luonnonlisen kielen jäsennin" (A finite state parser of natural language). Licentiate (pre-doctoral) thesis. Department of Computer Science, University of Helsinki.
- (Taylor & Kroch, 1994) Taylor, A.; Kroch, A. *The Penn-Helsinki Parsed Corpus of Middle English*. 1994. (Disponível em <http://www.ling.upenn.edu/mideng>)
- (van Halteren, 1998) van Halteren, H.; Zavrel, J.; Daelemans, W. Improving Data Driven Wordclass Tagging by System Combination. *COLING-ACL 1998*. Montreal, Canada, 1998. P. 491-497
- (Vapnik, 1995) Vapnik, Vladimir N. *The Nature of Statistical Learning Theory*. Springer, 1995.
- (Villavicencio et al., 1995) Villavicencio, A. et al. Part-of-Speech Tagging for Portuguese Texts. In: *Anais do Simpósio Brasileiro de Inteligência Artificial (SBIA '95)*. 1995.
- (Villavicencio et al., 1996) Villavicencio, A. et al. Evaluating Part-of-Speech Taggers for the Portuguese Language. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Curitiba, 1996. p. 159-168.
- (Villavicencio, 1995) Villavicencio, A. *Avaliando um rotulador estatístico de categorias morfo-sintáticas para o português*. Rio Grande do Sul, 1995. Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul.
- (Viterbi, 1967) Viterbi, A. J. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, p 260-269. 1967.
- (Voutilainen & Tapanainen, 1993) Voutilainen, A. And Tapanainen, P. Ambiguity Resolution in a Reductionistic Parser. Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics. Utrecht. 394-403.
- (Voutilainen et al., 1992) Voutilainen, A., Heikkila, J. And Antilla, A. *Constraint Grammar of English: A Performance-Oriented Introduction*. Technical Report Publication No. 21, University of Helsinki, Department of General Linguistics, Helsinki. 1992.
- (Voutilainen, 1994) Voutilainen, A. *Three studies of grammar-based surface parsing of unrestricted English text*. (Doctoral dissertation.). Publications 24, Department of General Linguistics, University of Helsinki.
- (Voutilainen, 1995) Voutilainen, A. *A syntax-based part-of-speech analyser*. 1995. (Disponível em <http://xxx.lanl.gov/cmp-lg/9502012>)
- (Wilkens & Kupiec, 1996) Wilkens, M. And Kupiec, J. *Training Hidden Markov Models for Part of Speech Tagging – Revision 4*. 1996.
- (Disponível em <http://www.nxrc.xerox.com/grenoble/mltt/fsNLP/train.html>)
- (Witten & Frank, 2000) Witten, I. H.; Frank, Eibe. *Data mining: practical machine learning tools and techniques with Java implementations*. Academic Press, 2000.
- (Wynne, 1995) Wynne, Martin. *A post-editor's guide to Claws 5 Tagging*. UCREL. University of Lancaster, 1995. (Disponível em <http://www.comp.lancs.ac.uk/computing/users/paul/ucrel/claws/>)
- (Wynne, 1996) Wynne, Martin. *A post-editor's guide to Claws 7 Tagging*. UCREL. University of

- Mencionados na WEB. In: *Anais do III Encontro para o Processamento Computacional de Português Escrito e Falado (PROPOR'98)*. Porto Alegre, 1998. p. 38-47.
- (Pacheco et al., 1996) Pacheco, H.c. da Fonseca. et al. Uma nova abordagem para análise sintática do português. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Curitiba, 1996. p. 51-60.
- (Paiva et al., 1996) Paiva, Daniel da S. de.; Carvalho, Adriane M. B. R. e Wainer, Jacques. Um sistema para Processamento Distribuído de Linguagem Natural. In: *Anais do II Encontro para Processamento Computacional do Português Escrito e Falado*. Curitiba, 1996. p. 81-90.
- (Pinheiro, 1990) Pinheiro, João Batista Gonçalves Pinheiro. *Tópicos de Linguagem – Gramática – Análise Sintática – Teoria e Prática*. Atual Editora. 1990. 9ª edição.
- (Quinlan, 1996) Bagging, boosting, and c4.5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (p. 725-730). AAAI Press and the MIT Press. 1996.
- (Ramshaw & Marcus, 1994) Ramshaw, L; Marcus, M. *Exploring the Statistical Derivation of Transformational Rule Sequences for Part-of-Speech Tagging*. 1994. (Disponível em <http://xxx.lanl.gov/cmp-lg/cmp-lg/9406011>)
- (Ratnaparkhi, 1996) Ratnaparkhi, A. A Maximum Entropy Part-of-Speech Tagger. *Proceedings of the First Empirical Methods in Natural Language Processing Conference*. Philadelphia, Pa. 1996.
- (Redington et al., 1998) Redington, Martin; Chater, Nick; Finch, Steven. Distributional Information: A Powerful Cue for Acquiring Syntactic Categories. *Cognitive Science*, vol 22 (4), p. 425-469. 1999.
- (Rigau et al., 1997) Rigau, G.; Atserias, J.; Agirre, E. Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. In *Proceedings of ACL/EACL '97*, p. 48-55.
- (Rocha & Dillinger, 1996) Rocha, N. and Dillinger, M. Interpretação de estruturas sintáticas do português. In: *Anais da V Semana de Iniciação Científica*. Belo Horizonte: Universidade Federal de Minas Gerais, 1996.
- (Roth, 1998) Roth, D. Learning to Resolve Natural Language Ambiguities: A Unified Approach. In *Proceedings of AAAI'98*. p. 806-813.
- (Salton et al., 1990) Salton, G.; Zhao, Z.; Buckley, C. *A simple syntactic approach for the generation of indexing phrases*. Technical Report 90-1137, Department of Computer Science, Cornell University. 1990.
- (Santorini, 1990) Santorini, B. *Part of Speech Tagging Guidelines for the Penn Treebank Project*. 1990. (Disponível em <http://www.cis.upenn.edu/treebank/home.html>)
- (Schmid, 1994) Schmid, H. *Part-of-Speech Tagging with Neural Networks*. 1994. (Disponível em <http://xxx.lanl.gov/cmp-lg/>)
- (Schmid, 1995) Schmid, H. *Probabilistic Part-of-Speech Tagging Using Decision Trees*. 1995. (Disponível em <http://www.ims.uni-tübingen.de/Tools/DecisionTreeTagger.html>)
- (Schütze, 1995) Schütze, H. *Distributional Part-of-Speech Tagging*. 1995. (Disponível em <http://xxx.lanl.gov/cmp-lg/9503009>)
- (Sharkey, 1999) Sharkey, A.J.C. *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, London: Springer. 1999
- (Silva & Lima, 1996) Silva, J.L. Tavares da. e Lima, V.I. S. de. Algumas Considerações sobre

Lancaster, 1996. (Disponível em <http://www.comp.lancs.ac.uk/computing/users/paul/ucrel/claws/>)
(Young & Bloothooft, 1997) Young, S.; Bloothooft, G. *Corpus-Based Methods in Language and Speech Processing*. Kluwer Academic Publishers, 1997.

APÊNDICE A – NILC TAGSETS

A definição do conjunto de etiquetas que será utilizado na etiquetagem de um corpus é uma tarefa um tanto complexa, pois envolve conhecimento lingüístico e conhecimento sobre os fatores que influenciam no desempenho de um dado etiquetador. Alguns dos itens que devem ser considerados são: (a) o tamanho do conjunto de etiquetas e (b) a quantidade de informação lingüística necessária para realizar uma determinada tarefa, por exemplo a inclusão ou não de informações de número e gênero (EAGLES, 1996a).

No entanto, mesmo com uma elaboração cuidadosa do conjunto de etiquetas, pode haver a necessidade de redefinir-lo dado uma etiqueta que faltou, uma etiqueta desnecessária, etc. – detalhes que só são observados durante a etiquetagem manual do corpus. Foi o que aconteceu neste trabalho em que o conjunto de etiquetas foi reformulado quatro vezes, em uma primeira instância para sanar problemas de eficiência de treinamento do etiquetador TBL, em outra para cobrir novos casos que foram observados na etiquetagem manual do corpus.

O primeiro NILC tagset foi criado de acordo com as recomendações contidas em (EAGLES, 1996) para se incluir atributos e valores para as 14 categorias principais do conjunto de etiquetas:

S (substantivo)	PP (preposição)	E (residual-palavras estrangeiras)
V (verbo)	C (conjunção)	SE (residual-símbolos especiais)
AJ (adjetivo)	N (numeral)	FM (residual-fórmulas matemáticas)
PR (pronome)	AR (artigo)	símbolo (pontuação)
AV (advérbio)	I (interjeição)	SI (siglas)
		AB (abreviaturas)

As recomendações levaram a cada etiqueta de uma palavra ser formada pelos valores, em seqüência, de cada atributo relativo a cada categoria do conjunto de etiquetas. Note que, separando cada valor usa-se um ponto, por exemplo, "mesa" seria macada com S.C.F.S.N; "carrinhos" com S.C.M.P.D; e "personagem" com S.C.2G.S.N, dados os atributos e valores da categoria substantivo:

Tipo	Comum (C)	Próprio (P)		
Gênero	Masculino (M)	Feminino (F)	2G (2G)	
Número	Singular (S)	Plural (P)	2N (2N)	Invariável (I)
Grau	Aumentativo (A)	Diminutivo (D)	Neutro (N)	

Na segunda versão do NILC tagset, o conjunto de etiquetas foi reduzido, pois não havia necessidade do aprendizado de muitos dos atributos das categorias principais. Uma vez que o etiquetador consegue descobrir a categoria, o texto poderia ser pós-processado com a ajuda do léxico do NILC, preenchendo-se os valores para os atributos restantes. Foi ainda na segunda versão que as etiquetas para siglas e abreviaturas desapareceram como categorias principais sendo renomeadas com a categoria de substantivo; que criamos etiquetas para os verbos "ter" e "ser"; e que começamos a usar símbolos para marcar contrações, ênclises e mesóclises. Esta segunda versão parece-se muito com o conjunto de etiquetas utilizado por Marques & Lopes (1996) na etiquetagem de textos em português continental. Na terceira versão do Nilc Tagset houve mudança nas etiquetas para pronomes; os verbos começaram a ser avaliados conforme a transitividade; e além das duas etiquetas para os verbos "ter" e "ser", passamos a ter também etiquetas para os verbos "ir", "estar" e "haver", e também uma etiqueta para palavras residuais. Na quarta versão excluímos as etiquetas para os verbos "ter", "ser", "ir", "estar" e "haver" e

criamos uma etiqueta para verbos auxiliares. E, finalmente, na quinta versão (atual) foram criadas as etiquetas para locuções e para palavra denotativa, item de lista e os coloquialismos e regionalismos passaram a fazer parte da etiqueta residual. Este apêndice traz as cinco versões do NILC tagset.

A1 – NILC TAGSET (Versão 1)

1. SUBSTANTIVO (S)

Atributos e Valores

Tipo	Comum (C)	Próprio (P)		
Gênero	Masculino (M)	Feminino (F)	2G (2G)	
Número	Singular (S)	Plural (P)	2N (2N)	Invariável (I)
Grau	Aumentativo (A)	Diminutivo (D)	Neutro (N)	

Exemplos: mesa /SCFSN
 carrinhos /SCMPD
 personagem /SC2GSN
 lápis /SCMIN
 Brasil /SPMSN

2. ADJETIVO (AJ)

Gênero	Masculino (M)	Feminino (F)	2G (2G)	
Número	Singular (S)	Plural (P)	2N (2N)	
Grau	Aumentativo (A)	Diminutivo (D)	Neutro (N)	Superlativo (S)

Exemplos: belo /AJMSN
 amicíssimas /AJFPS

3. ARTIGO (AR)

Tipo	Definido (D)	Indefinido (I)
Gênero	Masculino (M)	Feminino (F)
Número	Singular (S)	Plural (P)

Exemplos: os /ARDMP
 uma /ARIFS

4. PREPOSIÇÃO (PP)

Exemplos: desse /PP; de /PP; ao /PP

5. CONJUNÇÃO (C)

Tipo	Coordenativa	Subordinativa
Complemento	Adit (AT) Adv (AV) Alter (AL) Concl (CL) Expl (E)	Integ (I) Caus (CA) Comp (CM) Conc (CC) Cond (CD) Cons (CS) Fin (F) Temp (T) Propor (P) Confor (CF)

Exemplos: e /C[AT|AV]
que /C[[AT|AL][I|CA|CM|F|CC]]

6. NUMERAL (N)

Tipo	Cardinal (C)	Ordinal (O)	Multiplicativo (M)	Fracionário(F)	Frac-Ord. (FO)
Gênero	Masculino (M)	Feminino (F)	2G (2G)	Invariável (I)	
Número	Singular (S)	Plural (P)	2N (2N)	Invariável (I)	
Função	Pronome (P)	Adjetivo (A)			

Exemplos: o quarto /NFOMSA poder
duplo /NMMSA sentido
o primeiro /NOMSP foi meu pai, o segundo....

7. INTERJEIÇÃO (I)

Exemplo: Ei, Oh, Ah, Ui

8. PRONOME (PR)

Tipo	Retó (RT)	Obl. Átono (OA)	Obl. Tônico (OT)	Possessivo (P)	Indefinido (ID)	Interrogat. (IT)	Relativo (RL)	Reflexivo (RF)	Tratamento (T)	Demonstr. (D)
Gênero	Masculino (M)	Feminino (F)	2G (2G)	Invariável (I)						
Número	Singular (S)	Plural (P)	2N (2N)	Invariável (I)						
Pessoa	1ª. Sing. (1S)	2ª. Sing. (2S)	3ª. Sing. (3S)	1ª. Plural (1P)	2ª. Plural (2P)	3ª. Sing/ Plu (3SP)	Neutro (N)			

Exemplos: desse /PRDMS2S
os /PR[D|OA]MP3P livros estão aqui
o /PRDMSN que eu gostaria de dizer, já o /PRDMSN disse.

9. VERBO (V)

Predicação	Intransitivo (I)	Trans. Direto (TD)	Trans. Indir. (TI)	Bitransitivo (BI)	Ligaçāo (L)	Auxiliar (A)	Pronominal (P)				
Formas Nominais	Inf. Pessoa 1 (IP)	Gerúndio (G)	Particípio (P)	Inf. Impessoal??? (II)							
Tempo	Presente (PS)	Pret. Imper. (PI)	Pret. Perf. (PP)	Pret+Perf. (PMP)	Fut. Pres. (FPS)	Fut. Prét. (FP)	Pres. Subj (PSU)	Pret. Imp. Subj (PIS)	Fut. Subj (FS)	Imp. Afirm (IA)	
Pessoa	1º.Sing. (1S)	2º. Sing. (2S)	3º. Sing. (3S)	1º. Plural (1P)	2º. Plural (2P)	3º. Plural (3P)					
Gênero	Masculino (M)	Feminino (F)	2G (2G)								
Número	Singular (S)	Plural (P)	2N (2N)								

Exemplos:

10. ADVÉRBIO (AV)

Tipo	Circ-Lugar (CL)	Circ-Tempo (CT)	Circ-Modo (CM)	Negação (N)	Dúvida (D)	Interrogativo (I)	Afirmativo (A)	Int-Lugar (IL)	Int-Tempo (IT)	Int-Modo (IM)	Int-Causa (IC)
Grau	Aumentativo (A)		Diminutivo (D)		Superlativo (S)			Neutro (N)			

11. SIGLAS (SI)

Ex. IBM CACM ICAI ICMC USP

12. ABREVIATURAS (AB)

EX. ex. i.e. exmo. Ilmo. Sra.

13. PONTUAÇÃO (o próprio símbolo é o tag)

. ! ? ... ; , : () " — [] { }

14. RESIDUAL

É atribuída a palavras que estão fora das classes tradicionais, embora ocorram com frequência. Ex: palavras estrangeiras, fórmulas matemáticas (quando aparecem sem espaço em branco, como em X/21, 2+3=5), símbolos especiais, como @, R\$, \$, %, #, +, -, =, ^, ~, <, >, /, \

É dividida em 3 subcategorias:

(a) PALAVRAS ESTRANGEIRAS (E)

Não se trata de estrangeirismos (ex. flat, shopping, abajour, pizza), ou seja, não são palavras já incorporadas ao Português.

Ex. home

Love

(b) SÍMBOLOS ESPECIAIS (SE)

São os símbolos com exceção daqueles da classe 14. Pontuação:

@, R\$, \$, %, #, +, -, =, ^, ~, <, >, /, \ e todos os demais que podem ser inseridos dos alfabetos especiais (Symbol, etc.)

(c) FÓRMULAS MATEMÁTICAS (FM)

São expressões, sem espaço entre os símbolos, que não constituem palavras do português.

Ex. f(x)

x+y (observe que x + y – com espaço – seria classificado como x/S +/SE y/S

35/4

10%

A2 – NILC tagset (Versão2)

É composto por 49 etiquetas sem contar as etiquetas compostas pela utilização dos operadores de contrações, mesóclises e ênclises, sendo que destas 15 são utilizadas para pontuação.

São utilizados dois operadores²⁶, um para contrações (da, desses, destes, daquela) e ênclises (dar-lhe, dei-lhe, dei-o) – +, e outro para mesóclises (dar-lhe-ei, tirá-lo-ei) – !. Utilizando estes operadores, alguns exemplos de etiquetagem seriam: “do” = PREP+ART, “dar-lhe” = VINF+ART e, “dar-lhe-ei” = VINF!PPOA.

Categoría morfossintática	Etiqueta
Adjetivo	ADJ
Adverbio	ADV
Artigo	ART
Numeral Cardinal	NC
Numeral Ordinal	ORD
Outros Numerais	NO
Substantivo Comum	N
Nome Próprio + Siglas	NP
Conjunção Coordenativa	CONCOORD

²⁶ Estes operadores são utilizados nas versões 2, 3, 4 e 5 do NILC tagset.

Conjunção Subordinativa	CONJSUB
Pronome Demonstrativo	PD
Pronome Indefinido	PIND
Pronome Obliquo Átono	PPOA
Pronome Pessoal Caso Reto	PPR
Pronome Possessivo	PPS
Pronome Relativo	PR
Pronome Obliquo Tônico	PPOT
Pronome Interrogativo	PINT
Pronome Reflexivo	PREF
Pronome Tratamento	PTRA
Preposição	PREP
Verbo no Gerúndio	VGER
Verbo no Infinitivo	VINF
Verbo no Partípicio Passado	VPP
Verbo Ser no Gerúndio	VGERSER
Verbo Ser no Infinitivo	VINFSER
Verbo Ser no Partípicio Passado	VPPSER
Verbo Ter no Gerúndio	VGERTER
Verbo Ter no Infinitivo	VINFTER
Verbo Ter no Partípicio Passado	VPPTER
Verbo Ser (outras formas)	VSER
Verbo Ter (outras formas)	VTER
Verbo (outras formas)	V
Interjeição	I
.	.
:	:
—	—
((
,	,
!	!
?	?
...	...
))
“	“
[[
]]
{	{
}	}
‘	‘

A3 – NILC tagset (Versão 3)

Mesmo com as alterações, o número de etiquetas permanece estável, isto é, 49 etiquetas sem contar as etiquetas compostas pela utilização dos operadores de contrações, mesóclises e ênclises, sendo que destas 15 são utilizadas para pontuação.

Categoría morfossintática	Etiqueta
Adjetivo	ADJ
Advérbio	ADV
Artigo	ART
Numeral Cardinal	NC

Numeral Ordinal	ORD
Outros Numerais	NO
Substantivo Comum	N
Nome Próprio + Siglas	NP
Conjunção Coordenativa	CONCOORD
Conjunção Subordinativa	CONJSUB
Pronome Demonstrativo	PD
Pronome Indefinido	PIND
Pronome Oblíquo Átono	PPOA
Pronome Pessoal Caso Reto	PPR
Pronome Possessivo	PPS
Pronome Relativo	PR
Pronome Oblíquo Tônico	PPOT
Pronome Interrogativo	PINT
Pronome Apassivador	PAPASS
Pronome de Realce	PREAL
Pronome Tratamento	PTRA
Preposição	PREP
Verbo Ir	VIR
Verbo Ser	VSER
Verbo Estar	VESTAR
Verbo Ter	VTER
Verbo Haver	VHAVER
Verbo de Ligação	VLIG
Verbo Intransitivo	VINT
Verbo Transitivo Direto	VTD
Verbo Transitivo Indireto	VTI
Verbo Bitransitivo	VBI
Interjeição	I
Palavras ou Símbolos Residuais	RES ²⁷
.	.
:	:
—	—
((
,	,
!	!
?	?
...	...
))
“	”
[[
]]
{	{
}	}
‘	‘

A4 – NILC tagset (Versão 4)

²⁷ A etiqueta para palavras e símbolos residuais – RES – volta a ser utilizada e é composta pelas mesmas partes que a compunham na primeira versão do NILC tagset e permanece com a mesma estrutura até a versão 4 do NILC tagset.

Nesta versão do NILC tagset o número de etiquetas cai para 45 etiquetas, sem contar as etiquetas compostas pela utilização dos operadores de contrações, mesóclises e ênclises, sendo que destas 15 são de pontuação.

Categoría morfossintática	Etiqueta
Adjetivo	ADJ
Advérbio	ADV
Artigo	ART
Numerar Cardinal	NC
Numerar Ordinal	ORD
Outros Numerais	NO
Substantivo Comum	N
Nome Próprio + Siglas	NP
Conjunção Coordenativa	CONCOORD
Conjunção Subordinativa	CONJSUB
Pronome Demonstrativo	PD
Pronome Indefinido	PIND
Pronome Oblíquo Átono	PPOA
Pronome Pessoal Caso Reto	PPR
Pronome Possessivo	PPS
Pronome Relativo	PR
Pronome Oblíquo Tônico	PPOT
Pronome Interrogativo	PINT
Pronome Apassivador	PAPASS
Pronome de Realce	PREAL
Pronome Tratamento	PTRA
Preposição	PREP
Verbo Auxiliar	VAUX
Verbo de Ligação	VLIG
Verbo Intransitivo	VINT
Verbo Transitivo Direto	VTD
Verbo Transitivo Indireto	VTI
Verbo Bitransitivo	VBI
Interjeição	I
Palavras ou Símbolos Residuais	RES
.	.
:	:
-	-
((
!	!
?	?
...	...
))
"	"
[[
]]
{	{
}	}
:	:
,	,

A5 – NILC tagset (Versão 5 – Versão)

A versão atual do NILC tagset é um pouco maior que a versão 4, tem 51 etiquetas sem contar as etiquetas compostas pela utilização dos operadores de contrações, mesóclises e ênclises, sendo que 15 são de pontuação. Nela, a etiqueta para palavras e símbolos residuais – RES – passa a ser dividida em 4 subcategorias, as três das versões 1, 3 e 4 mais uma categoria para coloquialismos e regionalismos. Coloquialismos são expressões de estilo de linguagem informal (Por exemplo: Cê vai aonde?). Já os regionalismos são expressões próprias de uma região e regiões (Por exemplo: O bichinho venha aqui).

Categoría morfossintática	Etiqueta
Adjetivo	ADJ
Advérbio	ADV
Artigo	ART
Numeral Cardinal	NC
Numeral Ordinal	ORD
Outros Numerais	NO
Substantivo Comum	N
Nome Próprio + Siglas	NP
Conjunção Coordenativa	CONCOORD
Conjunção Subordinativa	CONJSUB
Pronome Demonstrativo	PD
Pronome Indefinido	PIND
Pronome Oblíquo Átono	PPOA
Pronome Pessoal Caso Reto	PPR
Pronome Possessivo	PPS
Pronome Relativo	PR
Pronome Oblíquo Tônico	PPOT
Pronome Interrogativo	PINT
Pronome Apassivador	PAPASS
Pronome de Realce	PREAL
Pronome Tratamento	PTRA
Preposição	PREP
Verbo Auxiliar	VAUX
Verbo de Ligação	VLIG
Verbo Intransitivo	VINT
Verbo Transitivo Direto	VTD
Verbo Transitivo Indireto	VTI
Verbo Bitransitivo	VBI
Interjeição	I
Locução Adverbial	LADV
Locução Conjuncional	LCONJ
Locução Prepositiva	LPREP
Locução Pronominal	LP
Locução Denotativa	LDEN
Palavra Denotativa	PDEN
Palavras ou Símbolos Residuais	RES
Ítem de lista	IL
.	.

:	:
-	-
((
!	!
?	?
...	...
))
"	"
[[
]]
{	{
}	}
'	'
,	,

APÊNDICE B – MANUAL DOS ETIQUETADORES

Este apêndice mostra quais são as ferramentas utilizadas e arquivos utilizados/gerados no treinamento e etiquetagem feitos pelos etiquetadores: TreeTagger, TBL, MXPOST e X.

B1 - TreeTagger

Para treinar o TreeTagger para uma nova língua, deve-se utilizar o comando:

train-tree-tagger [-st] [-cl] [-dtgl] [-ccw] [-atg] <léxico> <classes abertas> <arquivo de treinamento> <arquivo modelo>

Em que:

léxico é o arquivo que contém o léxico que será utilizado. Cada linha deve corresponder a uma palavra e um caractere de tabulação seguida dos possíveis pares de etiquetas/formas canônicas. As etiquetas e formas canônicas são separadas por espaços em branco. Sendo que não é obrigatório o uso da forma canônica, para tanto basta ter um hifen “-” onde estaria a forma canônica. Como mostrado no exemplo:

a ART PREP PPOA a
ou
a ART PREP PPOA -
menina N ADJ -
...

classes abertas é o arquivo que contém as etiquetas de classes abertas que poderão ser utilizadas pelo etiquetador, isto é possíveis etiquetas para palavras desconhecidas. Este arquivo deve conter apenas as etiquetas separadas por um espaço em branco. Por exemplo:

VTD VTI VINT VBI VAUX VLIG ADV N NP

arquivo de treinamento é o arquivo que contém os dados que serão utilizados no treinamento e também deve conter apenas uma palavra/símbolo de pontuação por linha seguidos por um caractere de tabulação e a etiqueta. Como mostra o exemplo abaixo:

A ART
menina N
é VLIG
bonita ADJ
.

arquivo modelo é o arquivo onde o TreeTagger armazenará o modelo gerado por ele neste treinamento.

Existem também alguns parâmetros adicionais que podem ser utilizados no treinamento:

-st <etiqueta sent>: serve para indicar quais são as etiquetas utilizadas para pontuações que indiquem final de período – ... ? ! . para os casos em que não é utilizada a etiqueta SENT que é a etiqueta default de pontuações que indicam final de período do TreeTagger.

-cl <tamanho do contexto>: indica o número de palavras anteriores que serão utilizadas como contexto. O valor default do TreeTagger para este parâmetro é 2, o que corresponde a um unígrafo.

-dtg <ganho mínimo da árvore de decisão>: indica um limiar para o qual o último nó da árvore de decisão deve ser deletado caso o ganho de informação seja menor. O valor default é 0,7.

-atg <ganho da árvore de afixo>: indica um limiar para o qual o último nó da árvore de afixo deve ser deletado caso o ganho de informação seja menor. O valor default é 0,15.

Feito isto, para etiquetar um texto, deve-se utilizar o comando:

tree-tagger [-token] [-lemma] [-sgml] [-proto] [-base] <arquivo de parâmetro> <arquivo de entrada> <arquivo de saída>

Em que:

arquivo de parâmetro é o arquivo onde está o modelo criado no treinamento, é o arquivo de saída do treinamento.

arquivo de entrada é o arquivo que deverá ser etiquetado. Deve conter apenas uma palavra/símbolo de pontuação por linha. Um arquivo válido, estaria no seguinte formato:

Ele
mora
em

arquivo de saída é o arquivo onde o etiquetador deverá armazenar o texto etiquetado.

Existem cinco parâmetros adicionais que podem ser utilizados na etiquetagem:

- token : faz com que o arquivo onde é armazenado o resultado da etiquetagem, apresente além das etiquetas suas palavras correspondentes.
- lemma : além da palavra caracter de tabulação etiqueta o arquivo conterá também a forma canônica de cada palavra.
- sgml : no caso do texto submetido ao etiquetador conter etiquetas SGML esta opção ignoratudo que aparecer entre '<' e '>'.
- proto : com esta opção o etiquetador gera também um arquivo chamado "lexicon-protocol.txt" que contém informações sobre o grau de ambigüidade e outras possíveis etiquetas de cada palavra.
- base : esta opção faz com que o etiquetador utilize apenas as informações lexicais na etiquetagem.

A Figura 31 mostra o fluxo dos dados no treinamento e na etiquetagem, as ferramentas que estão em laranja fazem parte do módulo X e são explicadas no Apêndice D.

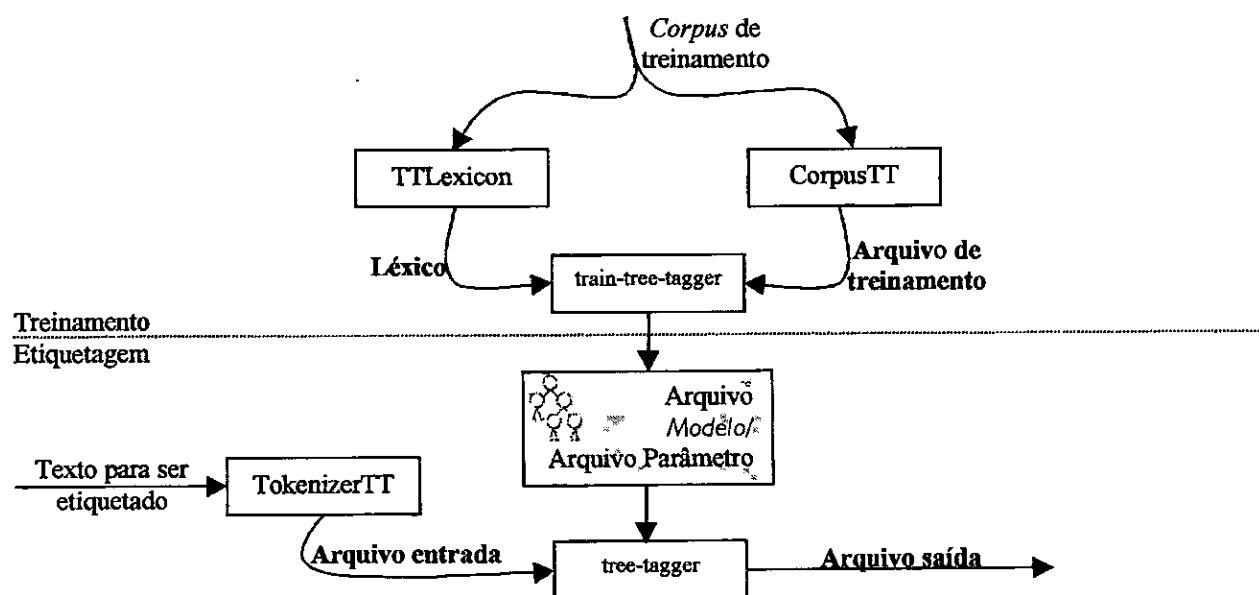


Figura 38 - Fluxo de dados no treinamento e etiquetagem com o etiquetador TreeTagger

B2 - TBL

O treinamento com o etiquetador TBL consiste de duas etapas: aprender regras para etiquetagem de palavras desconhecidas e aprender regras contextuais.

No treinamento para aprendizado de regras para palavras desconhecidas utiliza-se o comando:

unknown-lexical-learn.prl BIGWORDLIST SMALLWORDTAGLIST BIGBIGRAMLIST n LEXRULEOUTFILE
Em que:

BIGWORDLIST é um arquivo com todas as palavras/símbolos que estão presentes no corpus em ordem decrescente de freqüência.

de
a
o
e
que
do

SMALLWORDTAGLIST é um arquivo no formato - palavra etiqueta freqüência - que lista o número de vezes que uma palavra aparece com uma dada etiqueta no corpus.

, , 3093
. . 1787
de PREP 1411
o ART 909
e CONJCOORD 879
a ART 806

BIGBIGRAMLIST é um arquivo com as bigramas que aparecem no corpus de treinamento.

, Porto
permite que
única ,
das freqüentes
dar ordens
as concorrências
preocupações afastaram

n é um número que indica que deverão ser utilizados apenas os bigramas onde pelo menos uma das palavras é uma das *n* mais frequentes no corpus.

LEXRULEOUTFILE é o arquivo onde serão armazenadas as regras aprendidas.

No treinamento para aprendizado de regras contextuais utilizo-se o comando:

contextual-rule-learn TAGGED-CORPUS DUMMY-TAGGED-CORPUS \ CONTEXT-RULEFILE TRAINING.LEXICON

Em que:

TAGGED-CORPUS é o arquivo que contém o texto etiquetado manualmente, "tokenizado" e no formato de um período por linha

DUMMY-TAGGED-CORPUS é o arquivo formado pelo mesmo texto do arquivo TAGGED-CORPUS só que etiquetado pelo etiquetador inicial.

CONTEXTRULEFILE é o arquivo onde serão armazenadas as regras contextuais.

TRAININGLEXICON é o arquivo do léxico, que é formado pelas palavras e as possíveis etiquetas para cada palavra que tenham aparecido no corpus de treinamento.

rosto N

(Disponível em <http://www.folha.com.br/istoe> RES

vassouro-de-bruxa N

ortesanal ADJ

troca N LPREP

amarrada VTD ADJ

escrito ADJ N VINT VTD

Depois ADV LPREP LCONJ

Embora ADV CONJCOORD

O etiquetador TBL tem ferramentas auxiliares para gerar estes arquivos que são utilizados na treinamento e estão descritos no arquivo *readme* que acompanha o etiquetador.

Na etiquetagem é utilizado o comando:

tagger TRAININGLEXICON CORPUS BIGBIGRAMLISTS LEXICALRULEOUTFILE CONTEXTUALRULEFILE [opções]

Em que:

CORPUS é o nome do arquivo que será etiquetado.

Depois do nome de todos os arquivos, podem ser colocadas opções (Tabela 34).

Tabela 34 - Opções do comando tagger

-h	Help
-w wordlist	Provê um conjunto extra de palavras além das que estão no léxico
-i filename	Escreve o resultado intermediário do estado inicial em um arquivo
-s number	Para saber o número de linhas etiquetadas por tempo
-S	Usa apenas o etiquetador inicial
-F	Usa apenas o etiquetador final, ou seja o corpus tem de estar etiquetado

O etiquetador TBL provê ainda a possibilidade de aumentar-se a lista de bigramas e o léxico. Há também a possibilidade de se alterar manualmente as regras.

O fluxo de dados no treinamento do etiquetador TBL é mostrado na Figura 39.

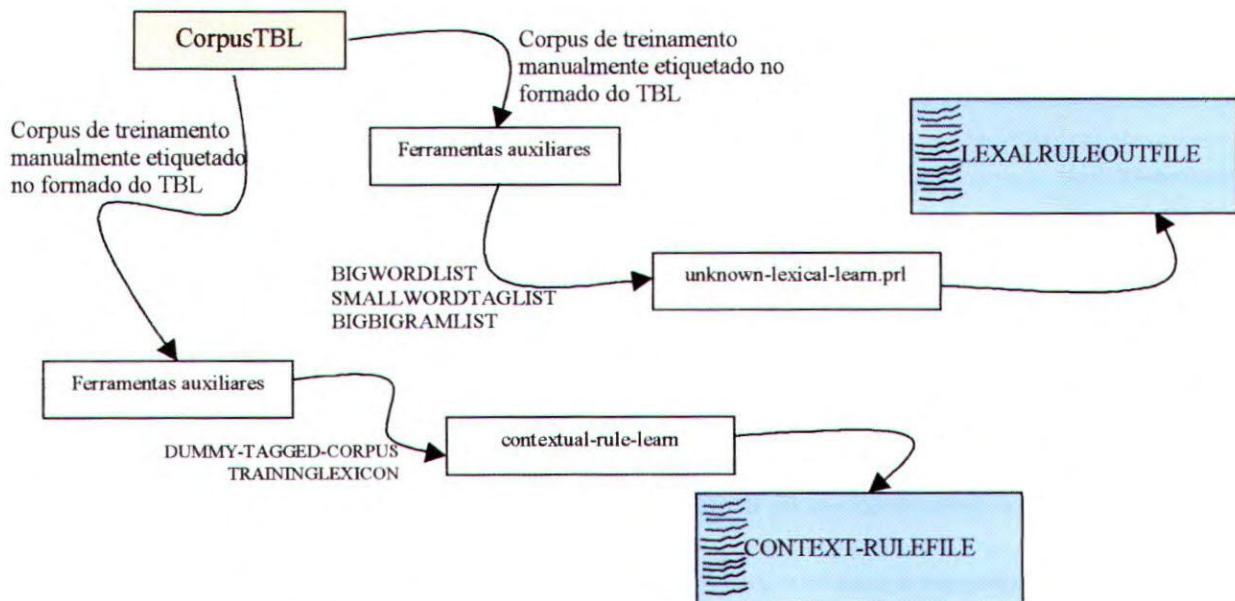


Figura 39 - Fluxo de dados no treinamento com o etiquetador TBL

B3 - MXPOST

O treinamento com o etiquetador MXPOST é feito através do comando:

```
trainmxpost <diretório> <corpus de treinamento>
```

Em que:

diretório: é o nome dado ao diretório onde estarão os arquivos do modelo gerado pelo etiquetador

corpus de treinamento: é o *corpus de treinamento* no formato *palavra_etiqueta*, com um período por linha

A etiquetagem do etiquetador MXPOST é feita através do comando:

```
mxpost diretório< texto > textosaída
```

Em que:

diretório é o nome do diretório onde está o modelo do etiquetador

texto é o texto que será etiquetado e que deve estar no formato: um período por linha

textosaída é o texto etiquetado pelo MXPOST

A Figura 34 mostra o fluxo de dados no treinamento e na etiquetagem do etiquetador MXPOST.

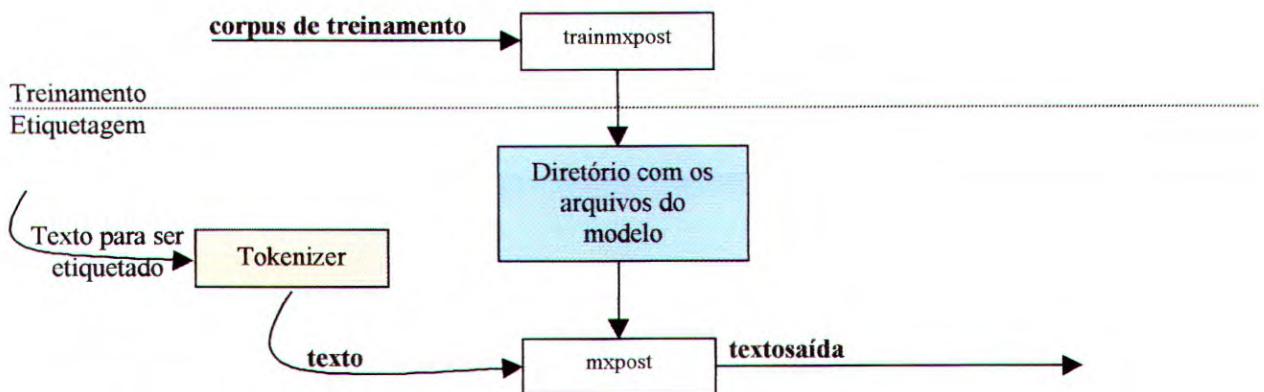


Figura 40 - Fluxo de dados no treinamento e etiquetagem com o etiquetador MXPOST

B4 - PoSiTagger

A treinamento com o etiquetador PoSiTagger é feita através do comando:

```
simbolico <texto> <textosaída>
```

Em que:

texto é o texto a ser etiquetado com um período por linha.

textosaída é o texto etiquetado no formato *palavra_etiqueta* com um período por linha.

A Figura 41 mostra o fluxo de dados no etiquetador PoSiTagger.



Figura 41 – Fluxo de dados no PoSiTagger

APÊNDICE C – REGRAS PARA DESAMBIGÜIZAÇÃO

GRAMATICAL

C1 - Regras para etiquetagem morfossintática do REGRA²⁸

<i>Regras Não-Lexicalizadas</i>
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VBI
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VESTAR
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VHAVER
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VINT
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VIR
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VSER
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VTD
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VTER
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaA = VTI
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaP = ADJ
Mude Etiqueta = ADJ para Etiqueta = ADV se EtiquetaP = ADV
Mude Etiqueta = ADJ para Etiqueta = N se EtiquetaA = , e EtiquetaP = , e Etiqueta2A = N
Mude Etiqueta = ADJ para Etiqueta = N se EtiquetaA = , e EtiquetaP = PREP e Etiqueta2A = N
Mude Etiqueta = ADJ para Etiqueta = N se EtiquetaA = , e EtiquetaP = PREP e Etiqueta2A = NP
Mude Etiqueta = ADJ para Etiqueta = N se EtiquetaA = PREP e EtiquetaP = ADJ
Mude Etiqueta = ADJ para Etiqueta = N se EtiquetaA = PREP e EtiquetaP = PREP

²⁸ Esta lista de regras foi obtida em julho de 1999.

Mude Etiqueta = ADJ para Etiqueta = N se EtiquetaA = PREP e EtiquetaP = PREP+ART
 Mude Etiqueta = ADJ para Etiqueta = N se EtiquetaP = PREP
 Mude Etiqueta = ADJ para Etiqueta = N se EtiquetaP = PREP+ART
 Mude Etiqueta = ADJ para Etiqueta = V se EtiquetaA = CONJSUB
 Mude Etiqueta = ADJ para Etiqueta = V se EtiquetaA = PPR
 Mude Etiqueta = ADV para Etiqueta = ADJ se EtiquetaA = VLIG
 Mude Etiqueta = ADV para Etiqueta = ADJ se EtiquetaP = N
 Mude Etiqueta = N para Etiqueta = V se EtiquetaA = CONJSUB
 Mude Etiqueta = N para Etiqueta = V se EtiquetaA = PPOA
 Mude Etiqueta = N para Etiqueta = V se EtiquetaA = PPOT
 Mude Etiqueta = N para Etiqueta = V se EtiquetaA = PPR
 Mude Etiqueta = N para Etiqueta = V se EtiquetaA = PPTRA
 Mude Etiqueta = NC para Etiqueta = N se EtiquetaA = , e EtiquetaP = V
 Mude Etiqueta = NC para Etiqueta = N se EtiquetaP = PREP
 Mude Etiqueta = NC para Etiqueta = N se EtiquetaP = PREP+ART
 Mude Etiqueta = NC para Etiqueta = N se EtiquetaP = VLIG
 Mude Etiqueta = V para Etiqueta = ADJ se EtiquetaA = VLIG
 Mude Etiqueta = V para Etiqueta = N se EtiquetaA = , e EtiquetaP = PREP
 Mude Etiqueta = V para Etiqueta = N se EtiquetaA = , e EtiquetaP = PREP+ART
 Mude Etiqueta = V para Etiqueta = N se EtiquetaA = ART
 Mude Etiqueta = V para Etiqueta = N se EtiquetaA = PPS
 Mude Etiqueta = V para Etiqueta = N se EtiquetaP = ADJ
 Mude Etiqueta = V para Etiqueta = N se EtiquetaP = V

Regras Lexicalizadas

Mude Etiqueta = N para Etiqueta = V se PalavraA = "quem"
 Se Palavra = "a" e PalavraP = mais então Etiqueta = ART
 Se Palavra = "a" e EtiquetaP = PAPASS então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PD então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PTND então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PINT então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PPOA então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PPOT então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PPR então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PR então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PREAL então Etiqueta = PREP
 Se Palavra = "a" e EtiquetaP = PTRa então Etiqueta = PREP
 Se Palavra = "claro" e PalavraA = ser então Etiqueta = ADV
 Se Palavra = "mesmo" e EtiquetaA = ADJ então Etiqueta = ADV
 Se Palavra = "muito" e EtiquetaP = ADJ então Etiqueta = ADV

Se Palavra = “pouco” e EtiquetaP = ADV então Etiqueta = ADV
Se Palavra = “pouco” e EtiquetaP = PREP então Etiqueta = ADV
Se Palavra = “quanto” e EtiquetaA = ADJ então Etiqueta = ADV
Se Palavra = “quanto” e EtiquetaP = ADJ então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VBI então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VESTAR então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VHAVER então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VINT então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VIR então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VSER então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VTD então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VTER então Etiqueta = ADV
Se Palavra = “rápido” e EtiquetaA = VTI então Etiqueta = ADV
Se Palavra = “só” e Etiqueta2A = VBI e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VESTAR e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VHAVER e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VINT e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VIR e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VLIG e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VSER e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VTD e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VTER e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “só” e Etiqueta2A = VTI e EtiquetaA = ADV então Etiqueta = ADJ
Se Palavra = “um” e PalavraA = “cada” então Etiqueta = N
Se Palavra = “um” e EtiquetaP = PREP então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VBI então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VESTAR então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VHAVER então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VINT então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VIR então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VLIG então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VSER então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VTD então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VTER então Etiqueta = N
Se Palavra = “um” e EtiquetaP = VTI então Etiqueta = N
Se Palavra = “uma” e PalavraA = “cada” então Etiqueta = N
Se Palavra = “uma” e EtiquetaP = “PREP” então Etiqueta = N
Se Palavra = “uma” e EtiquetaP = VBI então Etiqueta = N
Se Palavra = “uma” e EtiquetaP = VESTAR então Etiqueta = N

Se Palavra = "uma" e EtiquetaP = VHAVER então Etiqueta = N
Se Palavra = "uma" e EtiquetaP = VINT então Etiqueta = N
Se Palavra = "uma" e EtiquetaP = VIR então Etiqueta = N
Se Palavra = "uma" e EtiquetaP = VLIG então Etiqueta = N
Se Palavra = "uma" e EtiquetaP = VSER então Etiqueta = N
Se Palavra = "uma" e EtiquetaP = VTD então Etiqueta = N
Se Palavra = "uma" e EtiquetaP = VTER então Etiqueta = N
Se Palavra = "uma" e EtiquetaP = VTI então Etiqueta = N
Se Palavra = "uns" e PalavraP = "de" então Etiqueta = PIND
Se Palavra = "uns" e PalavraP = "dentre" então Etiqueta = PIND

Obs.: Etiqueta = Etiqueta da palavra que está sendo etiquetada;

EtiquetaA = Etiqueta da palavra anterior a que está sendo etiquetada;

Etiqueta2A = Etiqueta da palavra que está duas posições antes da que está sendo etiquetada;

EtiquetaP = Etiqueta da palavra seguinte a que está sendo etiquetada;

Etiqueta2P = Etiqueta da palavra que está duas posições depois da que está sendo etiquetada;

Palavra = Palavra que está sendo etiquetada;

PalavraA = Palavra anterior a que está sendo etiquetada;

PalavraP = Palavra que vem em seguida a que está sendo etiquetada;

Palavra2A = Palavra que está duas posições antes da que está sendo etiquetada;

Palavra2P = Palavra que está duas posições depois da que está sendo etiquetada.

C2 - Regras do etiquetador TBL

As regras apresentadas aqui são resultado do treinamento do etiquetador TBL com 80% do corpus de treinamento — 376 regras para palavras desconhecidas e 271 regras contextuais —, e estão representadas da mesma forma com que estão no etiquetador. O número que aparece em cada regra é a pontuação de cada regra dada pelo etiquetador. Os seis exemplos abaixo são exemplos de regras para palavras desconhecidas para o inglês, para que fique claro o formato utilizado pelo etiquetador TBL.

1) NN s fhassuf 1 NNS 3022.2549507603848724

É o mesmo que dizer: Mude a etiqueta de NN para NNS se a palavra tiver o sufixo s

2) Ed hassuf 2 VBN 1203.543544204078895

É o mesmo que dizer: Mude a etiqueta para VBN se o sufixo é "ed"

3) Ly addsuf 2 JJ 578.41553589321256368

É o mesmo que dizer: Mude a etiqueta para JJ se adicionando o sufixo "ly" o resultado é uma palavra

4) - char JJ 425.28095238095232844

É o mesmo que dizer: Mude a etiqueta para JJ se o caracter "-" aparecer na palavra

5) NN to fgoodright VB 224.95212906910632

É o mesmo que dizer: Mude a etiqueta de NN para VB se a palavra vier sempre a direita da palavra "to"

6) Un deletepref 2 JJ 67.245614035087726279

É o mesmo que dizer: Mude a etiqueta para JJ se deletando o prefixo "un" resultar em uma palavra.

C21 Regras para palavras desconhecidas

N o fgoodleft VTD 192.510261367627
u hassuf 1 VTD 132.94880952381
- char VTD+PPOA 130.733333333333
N ar fhassuf 2 VTD 128.921153846154
N do fhassuf 2 VTD 126.31829004329
N m fhassuf 1 VTD 111.126984126984
ente hassuf 4 ADV 99.333333333333
N l fhassuf 1 ADJ 79.968426018426
mais goodright ADJ 67.1399168651192
1 char NC 66
VTD o fgoodright N 58.822095104017
m addsuf 1 VTD 50.4225188604499
ada hassuf 3 ADJ 42.633333333333
N ica fhassuf 3 ADJ 37.7619047619048
ava hassuf 3 VTD 37.5
anos goodleft NC 35.4494172494173
N ais fhassuf 3 ADJ 34.68181818182
N adas fhassuf 4 ADJ 33.3095238095238
N ico fhassuf 3 ADJ 27.3666666666667
dos hassuf 3 ADJ 25.6380952380952
ADJ de fgoodright N 35.8223208971185
uma goodright N 25.4985514485514
N não fgoodright VTD 22.5646207723231
0 char NC 22
um goodright N 21.059188034188
muito goodright ADJ 20.7765656565657
N oso fhassuf 3 ADJ 20
N icos fhassuf 4 ADJ 19.0666666666667
N ir fhassuf 2 VTD 18.3888888888889
osa hassuf 3 ADJ 18
se goodright VTD 17.1666666666667
ei hassuf 2 VTD 16.333333333333
icas hassuf 4 ADJ 14
VTD da fgoodleft VTI 13.3666666666667
me goodright VTD 14.1397058823529
N ana fhassuf 3 ADJ 13
2 char NC 13
ida hassuf 3 ADJ 12.9444444444444
mos hassuf 3 VTD 12.5
tão goodright ADJ 12
VTD do fgoodright N 11.8333333333333
N ante fhassuf 4 ADJ 11.3333333333333

d addpref 1 PD 10.7826086956522
VTD s fdeletesuf 1 N 10.5
n deletepref 1 PREP+PD 9.51136363636364
ADV s faddsuf 1 ADJ 9.33333333333333
ADJ da fgoodright N 9.01709401709402
ADJ dos fgoodright N 8.93333333333333
VTD ao fgoodleft VTI 8.31929181929182
gem hassuf 3 N 8
veis hassuf 4 ADJ 8
7 char NC 8
d deletepref 1 PREP+PD 7.67000648403087
ADJ foi fgoodright VTD 7.66666666666667
rá hassuf 2 VTD 7.5
N er fhassuf 2 VTD 7.10714285714286
VTD no fgoodright N 7
VTD+PPOA da fgoodright N 7
osos hassuf 4 ADJ 7
VTD pod fhaspref 3 VAUX 6.66666666666667
ADJ o fgoodright N 6.32227994227994
vel hassuf 3 ADJ 7
VTD par fhaspref 3 VTI 6.25
VTD+PPOA o fgoodright N 6
asse hassuf 4 VTD 6
VTD da fgoodright N 6.78571428571429
VTD br fhaspref 2 VINT 6
fic haspref 3 VLIG 5.80952380952381
N deletepref 1 PREP+PD 5.75
N ária fhassuf 4 ADJ 5.33333333333333
N estão fgoodright ADJ 5.33333333333333
VTD à fgoodleft VTI 5.21666666666667
oco haspref 3 VINT 5.21428571428571
N está fgoodright ADJ 5.03333333333333
ú haspref 1 ADJ 5
anas hassuf 4 ADJ 5
osas hassuf 4 ADJ 5
VTD cado fhassuf 4 ADJ 5
sistema goodright ADJ 5
N quem fgoodright VTD 5
NC - fchar NP 5
VTD vo fhaspref 2 VINT 5
-lhe deletesuf 4 VTI+PPOA 5
NP as fgoodleft PREP 4.88179347826087
N iva fhassuf 3 ADJ 4.83333333333333
ADJ de fgoodright N 5.01010101010101
muito goodright ADJ 7.77656565656566
N ainda fgoodright ADJ 5.06592238171186
N estava fgoodright ADJ 4.77142857142857
mundo goodright ADJ 4.69871794871795
ADJ sua fgoodleft PREP 5.80220001614082
da addpref 2 PREP+ART 4.65152460289782
VTD surg fhaspref 4 VINT 4.5
três goodright N 4.4192925155149
ecia hassuf 4 VINT 4.33333333333333
ceu hassuf 3 VINT 4.33333333333333
VTD entr fhaspref 4 VTI 4.25
N ção faddsuf 3 VTD 4.03846153846154
VTD com fgoodright N 4.83653846153846
N coisa fgoodleft ADJ 4.64619883040936
ex- haspref 3 N 4

* goodleft N 4
ADJ aos fgoodright N 4
VTD do fgoodright N 4
N imo fhassuf 3 ADJ 4
peq haspref 3 ADJ 4
N ext fhaspref 3 ADJ 4
pado hassuf 4 ADJ 4
verm haspref 4 ADJ 4
N forma fgoodright ADJ 4
tara hassuf 4 VTD 4
VTD dor fhaspref 3 VINT 4
VTD mor fhaspref 3 VINT 4
VTD cho fhaspref 3 VINT 4
VTD go fhaspref 2 VTI 4
VTD lu fhaspref 2 VTI 4
8 char NC 4
VTD obr fhaspref 3 VBI 4
VTD+PPOA do fgoodleft VBI+PPOA 4
NC h fchar RES 4
PREP+PD m fchar PREP+ART 4
nos haspref 3 PPS 4
guém hassuf 4 PIND 4
possível goodleft VLIG 3.93955611441657
aqua addpref 3 PPR 3.9162363740677
alg addpref 3 ART 3.78288274832467
N ano fhassuf 3 ADJ 3.66666666666667
N Não fgoodright VTD 3.66666666666667
VTI mas fgoodright ADV 3.50128101269158
N eu fgoodright VTD 3.5
VTD+PPOA para fgoodleft VBI+PPOA 3.5
mili haspref 4 ADJ 3.333333333333
VTD can fhaspref 3 VINT 3.333333333333
N tempo fgoodleft ADJ 3.28571428571429
Out haspref 3 ADJ 3.16666666666667
ADJ á fchar N 3
VTD um fhassuf 2 N 3
téc haspref 3 N 3
VTD+PPOA Na fgoodright N 3
ADV nte fdeletesuf 3 ADJ 3
Alg deletepref 3 ADJ 3
VTI ado fhassuf 3 ADJ 3
N nse fhassuf 3 ADJ 3
pró haspref 3 ADJ 3
diá haspref 3 ADJ 3
fem haspref 3 ADJ 3
N ime fhaspref 3 ADJ 3
ersa hassuf 4 ADJ 3
áida hassuf 4 ADJ 3
sivo hassuf 4 ADJ 3
ânea hassuf 4 ADJ 3
VTD gado fhassuf 4 ADJ 3
VTD hado fhassuf 4 ADJ 3
paul haspref 4 ADJ 3
VTD ficou fgoodright ADJ 3
N ministro fgoodright ADJ 3
N vezes fgoodright ADJ 3
N seres fgoodright ADJ 3
N mão fgoodright ADJ 3
NP am fhassuf 2 VTD 3

ADJ ndo fdeletesuf 3 VTD 3
N nde fhassuf 3 VTD 3
N -lhe faddsuf 4 VTD 3
N mas fgoodright VTD 3
VTD na fgoodright N 3
Á char NP 3
VTD gri fhaspref 3 VINT 3
VTD fum fhaspref 3 VINT 3
VTD desa fhaspref 4 VINT 3
VTD olh fhaspref 3 VTI 3
ADJ ; fgoodright ADV 3.47020961494646
N mil fgoodleft NC 3
VTD divi fhaspref 4 VBI 3
VTD conv fhaspref 4 VBI 3
VTD+PPOA div fhaspref 3 VBI+PPOA 3
+ char RES 3
/ char RES 3
à char PREP+PD 3
Aqu deletepref 3 PD 3
RES , fgoodleft NO 3
N 'fchar PREP+N 3
N células fgoodleft ADJ 2.97619047619048
N o fgoodleft VTD 190.517283065565
am hassuf 2 VTD 135.41847826087
u hassuf 1 VTD 134.745512820513
N ar fhassuf 2 VTD 126.383333333333
- char VTD+PPOA 114.892857142857
N do fhassuf 2 VTD 114.721428571429
ente hassuf 4 ADV 93.6666666666667
mais goodright ADJ 73.974115942315
1 char NC 63
N 1 fhassuf 1 ADJ 59.8545454545455
se goodright VTD 52.2800379572119
VTD o fgoodright N 48.5952738715225
N da fhassuf 2 ADJ 43.7708708708709
anos goodleft NC 36.2255131964809
ava hassuf 3 VTD 34.3666666666667
N ica fhassuf 3 ADJ 30.5714285714286
N ados fhassuf 4 ADJ 30.1666666666667
N das fhassuf 3 ADJ 30.1564102564103
ADJ de fgoodright N 37.1204229543515
m addsuf 1 VTD 28.8705500648897
N ico fhassuf 3 ADJ 28.8333333333333
N is fhassuf 2 ADJ 27.7151515151515
uma goodright N 23.1136521860206
N muito fgoodright ADJ 22.2219387755102
N oso fhassuf 3 ADJ 21
0 char NC 21
osa hassuf 3 ADJ 20
N ir fhassuf 2 VTD 17.6666666666667
VTD ado fhassuf 3 ADJ 17.4666666666667
ADJ foi fgoodright VTD 23.1666666666667
N er fhassuf 2 VTD 16.0333333333333
N i fhassuf 1 VTD 16
n deletepref 1 PREP+PD 13.4464285714286
N ana fhassuf 3 ADJ 13
2 char NC 13
N não fgoodright VTD 12.3839918830239
VTD do fgoodright N 12.3794871794872

tão goodright ADJ 12
icas hassuf 4 ADJ 11.7142857142857
ção addsuf 3 VTD 11.4107142857143
VTD da fgoodleft VTI 11.3214285714286
d addpref 1 PD 11.2528985507246
um goodright N 10.9404761904762
ADJ da fgoodright N 12.721645021645
N ante fhassuf 4 ADJ 10.7333333333333
VTD à fgoodleft VTI 10.4188453159041
amos hassuf 4 VTD 10
ADV s faddsuf 1 ADJ 9.8333333333333
ADJ do fgoodright N 9.31020408163265
ADJ dos fgoodright N 9.11818181818182
icos hassuf 4 ADJ 9.01428571428571
osas hassuf 4 ADJ 9
rá hassuf 2 VTD 9
VTD+PPOA he fhassuf 2 VBI+PPOA 9
N deletepref 1 PREP+PD 8.5
VTD do fgoodleft VTI 8.44223602484472
me goodright VTD 8.28535353535354
ex- haspref 3 N 8
osos hassuf 4 ADJ 8
N rem fhassuf 3 VINT 8
8 char NC 8
Na goodright N 7.42857142857143
VTD par fhaspref 3 VTI 7.1833333333333
VTD ao fgoodleft VTI 7
sem hassuf 3 VTD 7
N sse fhassuf 3 VTD 7
ceu hassuf 3 VINT 7
ADJ pela fgoodleft VTD 6.4333333333333
ADJ havia fgoodright VTD 6.3333333333333
VTI s faddsuf 1 N 6.325
d deletepref 1 PREP+PD 6.03921356421356
emos hassuf 4 VTD 6
VTD+PPOA de fgoodleft VBI+PPOA 6
3 char NC 5.7
NC h fchar RES 6
N ária fhassuf 4 ADJ 5.66666666666667
N está fgoodright ADJ 5.66666666666667
N ivo fhassuf 3 ADJ 5.66666666666667
VTD da fgoodright N 5.61904761904762
ADJ no fgoodright N 5.6
fazer goodleft VAUX 5.42029686094904
N estava fgoodright ADJ 5.34883720930233
ADJ são fgoodright VTD 5.14166666666667
VTD ; fgoodleft VINT 5.03888888888889
coisa goodleft ADJ 5.1213468013468
VTD+PPOA ã fchar N 5
nea hassuf 3 ADJ 5
vel hassuf 3 ADJ 5
/ char RES 5
VTD+PPOA tor fhaspref 3 VLIG+PPOA 5
N ainda fgoodright ADJ 4.90681272509004
pró haspref 3 ADJ 4.85714285714286
ADJ os fgoodright N 4.85714285714286
sala goodleft PREP+ART 4.75108225108225
surg haspref 4 VINT 4.66666666666667
ADJ das fgoodright N 4.625

foram goodright VTD 4.5
N fin fhaspref 3 ADJ 4.454545454546
lhe goodright VTD 4.38611111111111
VTD ent fhaspref 3 VTI 4.7666666666667
duas goodright N 4.35
outras goodleft PREP 4.29241968516852
possível goodleft VLIG 4.21339808339808
N mulheres fgoodright ADJ 4.10164835164835
VTD+PPOA to fhassuf 2 N 4
hos hassuf 3 N 4
ú haspref 1 ADJ 4
bru haspref 3 ADJ 4
peq haspref 3 ADJ 4
osto hassuf 4 ADJ 4
ivas hassuf 4 ADJ 4
siva hassuf 4 ADJ 4
em deletesuf 2 VTD 4
N quem fgoodright VTD 4
Á char NP 4
NC - fchar NP 4
VTD resp fhaspref 4 VINT 4
VTD pass fhaspref 4 VTI 4.3051282051282
N m fdelete suf 1 VTI 4
VTD go fhaspref 2 VTI 4
VTD mor fhaspref 3 VTI 4
VTD dep fhaspref 3 VBI 4
VTD+PPOA para fgoodleft VBI+PPOA 4
+ char RES 4
PREP+PD m fchar PREP+ART 4
guém hassuf 4 PIND 4
nos haspref 3 PPS 4
aqua addpref 3 PPR 3.97872340425532
N ele fgoodright VINT 3.95
-lhe addsuf 4 VTD 3.9
ADJ pelas fgoodleft VTD 3.83333333333333
lon haspref 3 ADJ 3.7916666666667
alg addpref 3 ART 3.77646777597516
NP as fgoodleft PREP 3.767045454545
VTD tão fgoodleft VLIG 3.76189338883392
VTI mas fgoodright ADV 3.75919732441472
VTD contra fgoodleft VTI 3.7166666666667
fica haspref 4 VLIG 3.58333333333333
N cromossomos fgoodright ADJ 3.5
VTD va fhaspref 2 VINT 3.5
VTD exis fhaspref 4 VINT 3.5
VLIG . fgoodleft ADV 3.48133005556714
óide hassuf 4 ADJ 3.4
ADJ pelos fgoodleft VTD 3.33333333333333
ADJ sem fgoodright N 3.33333333333333
acab haspref 4 VTI 3.26666666666667
VTD trab fhaspref 4 VTI 3.26666666666667
PREP o fhassuf 1 CONJSUB 3.23214285714286
VTD oco fhaspref 3 VINT 3.21428571428571
sendo goodleft VLIG 3.1965591879385
VTD eram fgoodright ADV 3.17690058479532
NP : fgoodleft N 3.07142857142857
NP um fgoodleft VTD 4
pap haspref 3 N 3
ADJ dia fhaspref 3 N 3

ex- addpref 3 N 3
 VTD no fgoodright N 3
 VTD+PPOA do fhassuf 2 ADJ 3
 ADV nte fdeletesuf 3 ADJ 3
 VTI lar fhassuf 3 ADJ 3
 abe haspref 3 ADJ 3
 pio haspref 3 ADJ 3
 gro haspref 3 ADJ 3
 diá haspref 3 ADJ 3
 fem haspref 3 ADJ 3
 Out haspref 3 ADJ 3
 cido hassuf 4 ADJ 3
 mili haspref 4 ADJ 3
 paul haspref 4 ADJ 3
 bran haspref 4 ADJ 3
 N ministro fgoodright ADJ 3
 N vezes fgoodright ADJ 3
 N ar fgoodright ADJ 3
 N forma fgoodright ADJ 3
 N mundo fgoodright ADJ 3
 zado hassuf 4 VTD 3
 tara hassuf 4 VTD 3
 N sse faddsuf 3 VTD 3
 VTD au fhassuf 2 NP 3
 VTD S fchar VINT 3
 rol haspref 3 VINT 3
 VTD gri fhaspref 3 VINT 3
 dava hassuf 4 VINT 3
 viaj haspref 4 VINT 3
 reag haspref 4 VINT 3
 VTD apar fhaspref 4 VINT 3
 NP ei fhassuf 2 VTI 3
 VTD pene fhaspref 4 VTI 3
 VTD corr fhaspref 4 VTI 3
 7 char NC 3
 VTD obr fhaspref 3 VBI 3
 N d fhassuf 1 RES 3
 NC m fhassuf 1 NO 3
 VAUX as fgoodleft PREP 2.96741874349236

C22 Regras contextuais

N ADJ PREVTAG N
 PR CONJSUB PREVTAG VTD
 PIND ADJ NEXT1OR2TAG N
 ART PREP NEXT1OR2TAG ART
 VLIG VAUX NEXTTAG VTD
 PREP LPREP PREVTAG LPREP
 ADJ N PREVTAG PREP
 N ADJ NEXTTAG N
 ART PREP WDNEXTTAG a VTD
 VAUX VLIG NEXTTAG ART
 PREP+ART LPREP PREVTAG LPREP
 ART LP NEXTTAG PR
 PR LP PREVWD o
 N VTD PREVWD que
 ART PREP WDNEXTTAG a NC
 VINT VTD NEXT1OR2OR3TAG ART
 ADV LPREP NEXTWD de
 PREP LPREP PREVTAG LPREP

ART LPREP NEXTTAG LPREP
LADV ADJ PREV1OR2TAG N
I CONCOORD CURWD ora
NP VTD PREVTAG STAART
N ADJ PREVBIGRAM ADJ CONCOORD
VTD VTI NEXTWD de
ADJ VTD NEXTTAG ART
ADJ N SURROUNDTAG ART PREP+ART
VAUX VLIG NEXTTAG ADV
VTD VAUX NEXTTAG VAUX
ADJ N PREVTAG PREP+ART
ADJ VTD SURROUNDTAG VLIG PREP
VLIG VAUX NEXTTAG VTD
ART PREP NEXTTAG VINT
PR LCONJ PREVWD do
PREP+ART LCONJ NEXTTAG LCONJ
ART PPOA NEXTTAG VTD
PPOA CONJSUB NEXTTAG ART
PREP ART NEXTTAG N
LADV N PREVTAG NC
LDEN LCONJ PREV1OR2TAG STAART
N ADJ NEXTTAG N
N ADJ NEXTBIGRAM CONCOORD ADJ
ADJ VTD NEXTWD pelo
VAUX VLIG NEXTTAG PREP
VLIG VAUX NEXT1OR2TAG VBI
ADJ ADV CURWD mesmo
ADV ADJ WDNEXTTAG mesmo N
PREP LDEN NEXTTAG LDEN
ART PREP NEXTTAG PPR
ART PREP NEXTTAG PPOA
PREP LADV NEXTTAG LADV
VTD+PPOA N PREVTAG PREP+ART
PREP PPOA PREV1OR2TAG PR
ADJ N PREVBIGRAM STAART ART
ADJ VTD PREVTAG VAUX
PR CONJSUB NEXTWD os
ART PREP NEXTTAG VTI
ART PREP NEXTWD ser
VTI VINT NEXT1OR2OR3TAG ,
VLIG VAUX NEXTTAG VTI
VINT VTI NEXTBIGRAM PREP+ART N
PR LCONJ PREVWD de
PREP+ART LADV NEXTTAG LADV
ART LADV NEXTTAG LADV
ADV LCONJ NEXTTAG PR
PR LCONJ PREV1OR2TAG LCONJ
LP PREP NEXTTAG PR
CONJSUB PINT CURWD Que
VTD VINT NEXTTAG ,
VINT VTD PREV1OR2TAG PPOA
ADJ N PREVBIGRAM CONJSUB ART
ADJ N SURROUNDTAG ART PREP
VTD VINT NEXTBIGRAM ADV ,
VTD VINT SURROUNDTAG , PREP
VTD VTI NEXTWD sobre
VTD VAUX RBIGRAM tem que
VAUX VLIG NEXTTAG N
VTD VAUX WDNEXTTAG ter VTD

VLIG VAUX NEXTWD sendo
 VAUX VTD NEXTTAG ART
 PREP LPREP SURROUNDTAG , LPREP
 PREP LCONJ NEXTWD isso
 LPREP ADV WDAND2TAGAFT dentro N
 PREP LCONJ SURROUNDTAG N LCONJ
 PREP+ART LADV NEXTWD verdade
 N LADV PREVTAG LADV
 VTI VAUX RBIGRAM há de
 CONJSUB PREP PREVTAG VAUX
 PD LCONJ PREVTAG LCONJ
 ADJ N PREVBIGRAM , ADJ
 ADJ N PREVBIGRAM PREP ADJ
 N ADJ PREVTAG VLIG
 VTD VINT SURROUNDTAG N .
 VINT ADJ PREVBIGRAM PREP+ART N
 VTD VINT SURROUNDTAG PR PREP+ART
 N LADV WDPREVTAG PREP fora
 LADV ADV NEXT1OR2OR3TAG ,
 PR CONJSUB PREVTAG VLIG
 N VTD PREVBIGRAM " ,
 VINT VTD PREVTAG PIND
 VTD VAUX NEXTWD sido
 PR CONJSUB PREVTAG VTD
 PREP LADV NEXTWD dentro
 PR LCONJ PREVWD por
 PREP LCONJ RBIGRAM até que
 PR LCONJ PREVTAG LCONJ
 N ADV WDAND2TAGAFT meio N
 VTD+PPOA N PREVWD o
 VBI+PPOA VTD+PPOA PREV1OR2OR3TAG PREP
 VLIG LDEN RBIGRAM seja ,
 LPREP LADV PREVIOR2TAG LADV
 PPOA ART PREV1OR2TAG VTD
 PR LP PREVBIGRAM STAART LP
 VAUX N NEXTWD vivo
 CONJSUB LDEN PREVWD é
 NP PREP PREVTAG STAART
 PREP LADV NEXTTAG LADV
 LCONJ N PREV1OR2OR3TAG ART
 ADJ N PREVTAG ORD
 ADJ N PREVTAG -
 ADJ N PREVWD são
 ADJ N PREVBIGRAM ART PPS
 ADJ N CURWD moço
 VTD VINT NEXTTAG ?
 VTD VTI SURROUNDTAG NP PREP+ART
 VTD VTI SURROUNDTAG CONCOORD PREP
 VTI VINT NEXT1OR2TAG ADV
 VTD VBI NEXTBIGRAM ADV PREP+ART
 VTD ADJ CURWD conhecido
 ADJ N PREVWD velho
 N ADJ NEXTTAG N
 VTD N NEXTWD da
 N LADV RBIGRAM vez mais
 ADV LADV PREVTAG LADV
 ART PREP WDNEXTTAG a VTD+PPOA
 VTI VTD NEXTTAG ART
 N LADV LBIGRAM mesmo tempo

ADJ LADV NEXTTAG LADV
N VTD NEXTWD os
N LADV LBIGRAM em média
VTI VINT NEXT1OR2OR3TAG NC
N LADV WDPREVTAG PREP+PD caso
VAUX VLIG NEXTWD perdido
PR CONJSUB PREVTAG VTD+PPOA
VTD VAUX WDNEXTTAG tem VTD
N LADV WDPREVTAG ART princípio
VAUX VLIG WDNEXTTAG ser ADJ
PR CONJSUB PREVTAG VTI
PREP LADV NEXTTAG LADV
VINT VTI SURROUNDTAG N PREP
VBI VTD NEXT1OR2TAG .
VTD+PPOA VBI+PPOA NEXTWD do
VBI VTD NEXT1OR2OR3TAG ART
ADV LADV LBIGRAM a pouco
ART LADV NEXTTAG LADV
LREP ADV NEXT1OR2TAG ADV
LREP ADV WDPREVTAG N através
LREP PREP WDPREVTAG ADV de
PREP+ART LADV NEXTTAG LADV
ORD ADJ WDPREVTAG ART primeiros
PR PINT PREVBIGRAM STAART -
NC NP PREVTAG NP
LREP PREP+ART PREVTAG ADV
CONJSUB LCONJ RBIGRAM como se
LADV LREP NEXT1OR2TAG PREP+ART
PREP+ART PPOA NEXTTAG VINT
CONCOORD ADV CURWD embora
LCONJ LREP NEXT1OR2TAG N
PREP+ART LREP PREVTAG LREP
ORD PREP+ART PREVTAG ,
PTRA N CURWD senhor
ADJ ORD WDNEXTTAG segunda N
RES NO NEXTTAG ,
PREP+PD LADV NEXTTAG LADV
CONCOORD NP PREVTAG NP
CONCOORD LDEN NEXTTAG LDEN
ADJ N PREVTAG NC
ADJ N PREVWD ácidos
ADJ N RBIGRAM triptofano ,
ADJ N WDNEXTTAG ácido N
ADJ N PREV1OR2OR3TAG VBI
ADJ N NEXTBIGRAM - N
ADJ N SURROUNDTAG ART .
N ADJ PREV1OR2WD foi
N ADJ CURWD média
VTD VINT NEXTTAG ...
VTD VINT NEXTTAG LREP
VTD VINT NEXTTAG "
VTD VINT NEXTWD lá
VTD VINT LBIGRAM STAART Perguntou
VTD VINT WDPREVTAG N cai
VTD VINT PREV1OR2WD as
VTD VINT SURROUNDTAG PR PREP
VTD VINT SURROUNDTAG CONCOORD .
VTD VINT SURROUNDTAG ADV .
VTD VTI NEXTBIGRAM PREP VINT

VTD VTI SURROUNDTAG PREP PREP+ART
VTD VTI NEXT2WD usar
VTD VBI PREVBIGRAM VTD PPOA
VTD VBI PREV2WD estão
ADJ VTD PREVTAG PR
VTD ADJ SURROUNDTAG VLIG .
VINT VTD PREVWD nos
VINT VTD LBIGRAM , garante
VINT VTD NEXT1OR2WD por
VTD N PREVTAG ART
ART PREP NEXTWD caminho
VTD N CURWD scanner
VTI VTD NEXTTAG CONJSUB
ART PREP NEXTWD Istoé
N VTD PREVTAG NP
VTI VINT NEXT1OR2OR3TAG CONJSUB
ART PREP WDNEXTTAG a PD
VTI VTD CURWD chamado
ART PREP WDNEXTTAG a PIND
VTD VAUX LBIGRAM poderiam ter
VLIG VAUX NEXTWD classificados
N LADV PREVTAG LADV
ART PREP SURROUNDTAG N ADJ
VTD VAUX WDNEXTTAG teriam ADJ
VLIG VAUX WDPREVTAG N parece
PR CONJSUB PREVTAG VBI+PPOA
N LADV LBIGRAM de lado
PREP LADV NEXTTAG LADV
VAUX VLIG NEXTBIGRAM ADJ .
VTD VAUX CURWD podia
PR CONJSUB NEXTTAG PPS
VAUX VLIG WDAND2AFT ser .
PREP LADV NEXT1OR2WD vez
PR CONJSUB NEXT1OR2OR3TAG PR
PREP+ART LPREP LBIGRAM longo do
VINT VTI SURROUNDTAG PREP PREP
PREP+ART LPREP NEXTBIGRAM ADJ LPREP
VAUX VTD PREV1OR2TAG STAART
N LADV WDAND2TAGBFR VTD frente
ADJ PIND NEXTWD um
ADJ PIND WDAND2BFR , outros
ART PPOA NEXTTAG VAUX
ADV ADJ NEXTTAG .
VTD VLIG CURWD seriam
ADV LADV PREVWD nunca
VTI VTD PREV1OR2OR3TAG LADV
VTI VBI PREVWD foi
ADV ADJ NEXTWD gente
PREP ART PREVTAG PREP
LPREP ADV PREVTAG VINT
ADV ADJ PREVWD no
VINT ADJ PREV1OR2OR3TAG .
PREP+ART LADV NEXTWD tarde
PPOA CONJSUB PREVTAG ;
PREP LCONJ RBIGRAM por que
VTI N PREVTAG ART
VTD NP NEXT1OR2TAG (
LADV N LBIGRAM , conta
VLIG VINT NEXTTAG CONJSUB

PIND PR NEXT1OR2WD o
 N VTI PREV1OR2TAG LP
 VINT VAUX NEXTTAG VTD
 LP ART PREV1OR2OR3TAG ART
 LADV N PREV1OR2OR3TAG PPS
 PIND ADJ NEXT1OR2TAG N
 VLIG VTI WDNEXTTAG vai PREP
 ART NP PREVTAG PREP
 ADJ LPREP NEXTTAG LPREP
 VTD PPS LBIGRAM STAART Meu
 PREP+PD N NEXTTAG PREP+ART
 PPOA LCONJ PREVTAG LCONJ
 NP VLIG LBIGRAM STAART São
 NP PTR A CURWD Você
 VAUX ADJ PREV1OR2TAG VLIG
 NP PREP+ART LBIGRAM STAART Do
 ADJ N PREVTAG (

C3 Regras do etiquetador X

Regras iniciais para classes fechadas
Artigo – ART
O, a, os, as, um, uma, uns, umas
Pronome – Pessoal do Caso Reto – PPR
Eu, tu, ele, ela, nós, vós, eles, elas
Pronome – Pessoal Obliquo Átono – PPOA
Me, te, lhe, nos, lhes, vos, se
Pronome – Pessoal Obliquo Tônico – PPOT
Mim, comigo, ti, contigo, si, consigo, conosco, convosco
Pronome – Tratamento – PTR
Você, vocês, o Senhor, Sr., a Senhora, Sra., Vossa Santidade, V.S., Vossa Majestade, V. M., Vossas Majestades, VV. MM., Vossa Alteza, V. A., Vossas Altezas, VV. AA., Vossa Eminência, V. Ema., Vossas Eminências, V. Emas., Vossa Magnificência, Vossas Magnificências, Vossa Reverendíssima, V. Revma., Vossas, Reverendíssimas, V. Revmas., Vossa Excelência, V. Exa., Vossas Excelências, V. Exas., Vossa Senhoria, V. Sa., Vossas Senhorias, V. Sas.
Pronome – Interrogativos – PINT
Quem, qual, quando, Quanto, quantos, Quantas, onde
Pronome – Relativos – PR
Que, a qual, os quais, as quais, cujo, cujos, cuja, cujas
Pronome – Possessivo – PPS
Meu, minha, meus, minhas, teu, tua, teus, tuas, seu, sua, seus, suas, nosso, nossa, nossos, nossas, vosso, vossa, vossos, vossas, seu, sua, seus, suas
Pronome – Demonstrativo – PD
Este, esta, estes, estas, isto, esse, essa, esses, essas, isso, aquele, aquela, aqueles, aquelas, aquilo
Pronome – Indefinido – PIND
Alguém, ninguém, algo, outrem, tudo, nada, algum, nenhum, qualquer, todo, pouco, demais, tal, vários,ário, vária, várias, mais, cada
Preposição – PREP
Afora, ante, após, até, com, conforme, consoante, contra, de, desde, durante, em, entre, exceto, fora, malgrado, mediante, menos, para, per, perante, por, salvo, sem, sob, sobre, trás
Conjunção – Coordenativa – CONJCOORD
Consequentemente, e, já, logo, mas, nem, ora, ou, porém, portanto
Conjunção – Subordinativa – CONJSUB
Apenas, como, conquanto, embora, enquanto, mal, pois, porquanto, porque
Numeral – Cardinal – NC

Bilhão, bilhões, catorze, cem, cento, cinco, cinqüenta, dez, dezenove, dezesseis, dezessete, dezoito, dois, doze, duas, duzentas, duzentos, mil, milhão, milhões, nove, novecentas, novecentos, noventa, oitenta, oito, oitocentas, oitocentos, onze, Quarenta, quatorze, quatro, quatrocentas, Quatrocentos, quinhentas, quinhentos, Quinze, seis, seiscentas, seiscientos, sessenta, sete, setecentas, setecentos, setenta, três, treze, trezentas, trezentos, trilhão, trilhões, trinta, vinte, zero

Numeral – Ordinal – ORD

As palavras em azul podem ser numeral ordinal ou fracionário – serão fracionários quando vierem com outro número antes.

Duodécimo, duodécimos, enésima, enésimas, enésimo, enésimos, nongentésimo, nongentésimos, primeira, primeiras, primeiro, primeiros, segunda, segundas, segundo, segundos, setuagésima, setuagésimas, setuagésimo, setuagésimos, terceira, terceiras, terceiro, Terceiros, undécimo, undécimos, bilionésima, bilionésimas, bilionésimo, bilionésimos, centésima, centésimas, centésimo, centésimos, décima, décimas, décimo, décimos, ducentésima, ducentésimas, ducentésimo, ducentésimos, milésima, milésimas, milésimo, milésimos, milionésima, milionésimas, milionésimo, milionésimos, nona, nonagésima, nonagésimas, nonagésimo, nonagésimos, nonas, noningentésima, noningentésimas, noningentésimo, noningentésimos, nono, nonos, octingentésima, octingentésimas, octingentésimo, octingentésimos, octogésima, octogésimas, octogésimo, octogésimos, oitava, oitavas, oitavo, oitavos, quadragésima, Quadragésimas, quadragésimo, quadragésimos, quadringentésima, quadringentésimas, quadringentésimo, quadringentésimos, quarta, quartas, quarto, quartos, quinqüentésima, quinqüentésimas, quinqüentésimo, quinqüentésimos, quinquagésima, quinquagésimas, quinquagésimo, quinquagésimos, Quinta, quintas, quinto, quintos, septuagésima, septuagésimas, septuagésimo, septuagésimos, sétima, sétimas, sétimo, sétimos, setingentésima, setingentésimas, setingentésimo, setingentésimos, sexagésima, sexagésimas, sexagésimo, sexagésimos, sexcentésima, sexcentésimas, sexcentésimo, sexcentésimos, sexta, sextas, sexto, sextos, trecentésima, trecentésimas, trecentésimo, trecentésimos, trigésima, trigésimas, trigésimo, trigésimos, vigésima, vigésimas, vigésimo, vigésimos

Numeral – Outros Números – NO

Ambos, cêntuplo, dêcuplo, duodêcuplo, dupla, duplas, dúplex, díplices, duplo, duplos, meia, meias, meio, meios, nônuplo, óctuplo, quâdruplo, quintuplo, sétuplo, sétuplo, terça, terças, terço, Terços, triplo, undêcuplo

Interjeição – I

Estas expressões são interjeição quando aparecem exatamente desta forma seguidas de exclamação.

Ah, Ai De Mim, Ai, Alô, Alto Lá, Alto, Apoiado, Arreda, Atenção, Avante, Ave, Basta, Bico, Bis, Boa, Bravo, Calma, Céus, Chi, Chi, Coragem, Cuidado, Deus, Devagar, Diabo, Eh, Eia, Fora, Francamente, Hem, Hum, Hurra, Ih, Jesus(I), Meu Deus, Morra(I), Muito Bem, Ô, Ô, Oba, Oh, Olá, Olha Lá, Opa, Ora Bolas, Oxalá, Psit, Psiu, Puxa, Que Nada, Quê, Salve, Silêncio, Socorro, Ué, Uh, Ui, Upa, Valha-Me, Vamos, Viva

Regras iniciais para classes abertas

Adverbio – ADV

não, muito, alto, baixo, rápido, fundo, súbito, bem, depressa, devagar, ainda, agora, acima, abaixo, ligeiro, ligeira, sobremaneira, lá, acolá, amanhã, amiúde, longe, hoje, sempre, semelhante, diante, bastante, avante, doravante, defronte, distante, forte, leve, idem, ibidem, ontem, anteontem, também, além, aquém, sim, tão, então, adrede, talvez, grátils, depois, bis, jamais, ademais, antes, dantes, entrementes, algures, nenhures, alhures, aliás, deveras, arredor, derredor, porventura, sequer, assim, enfim, donde, mesmo, debaixo, assaz, tanto, quase, dentro, adentro, aí, acaso, afinal, meio, permeio, ali, aonde, aqui, cá, sobremodo, cedo, cedinho, debalde, decerto, detrás, tampouco, inda, quão

mente, inda	ADV – as palavras que terminam como os sufixos da lista a esquerda
-------------	--

Substantivo – Comum - N

mento, dor, dora, ança, ada, anda, fobia, cia, dia, grafia, fia, gia, lia, mia, nia, pia, tria, sai, tia, quia, xia, zia, ela, ola, ula, ama, oma, ura, eza, ice, dade, ase, ese, ise, ose, sse, ite, ol, gem, im, men, ção, ções, grafo, ifo, lho, eio, cópio, ismo, metro, mento, dez	N – as palavras que terminam com os sufixos da lista a esquerda.
---	--

Adjetivo – N

ivo, iva, ente, ais, ido, idos, forme, al, ável, ível, úvel, ico, esco, diço, ceo, neo, reo, seo, teo, veo, fero, gero, voro, oso, lento, issimo, érrimo	ADJ – as palavras que terminam com os sufixos da lista a esquerda.
--	--

Outras Regras iniciais	
Paravra	Etiqueta
Item de Lista – IL	
São considerados <u>índices</u> <u>as letras e números que vem seguidos de) ou -</u> . Note que sempre há espaço depois de), -	
Exemplos:	
a)	
1)	
1-	
i-	
ii)	
Palavra Denotativa – PDEN	
Eis, só, somente, exclusive, senão, inclusive, sobretudo	
Locuções Adverbiais – LADV	

à baila, à bala, a bandeiras despregadas, a bem dizer, a bordo, a cada passo, a capucha, a carga cerrada, a cavalinhas, a cavaleiro, a cavalo, a certa altura, a chucha calada, a chucha caladinha, a colação, a compita, a contento, a desora, a deus e a ventura, a deus misericórdia, à direita, à distância, a domicílio, a duras penas, a eito, à entrada, a escala vista, a escuta, a esmo, à espera, a espora fita, à esquerda, a facadas, a falsa fé, a farta, a fio, a fito, a flux, à francesa, à frente, a frio, a fundo, a furtapasso, a furta-passo, a furto, a galope, a gosto, a granel, ainda assim, ainda bem, ainda por cima, a lanço, a lápis, alguma vez, algumas vezes, a limpo, a lufa-lufa, a maior parte das vezes, a maioria da vezes, a mais, a mal, a mancheias, amanhã de manhã, amanhã de tarde, à mão, a mão tenente, a martelo, a máquina, a mata cavalos, a medo, a menos, à minha custa, à minha disposição, à minha espera, à minha vista, à minha volta, a monte, a nado, à noite, à noitinha, à nossa custa, à nossa disposição, à nossa espera, à nossa vista, à nossa volta, à nossa vontade, antes de ontem, antes pelo contrário, ao acaso, ao atar das feridas, ao certo, ao contrário, ao desbarato, ao deus dará, ao fim e ao cabo, ao fundo, ao invés, ao lado, ao largo, ao léu, a olhos vistos, ao longe, ao longo, ao menos, ao mesmo tempo, ao meu lado, ao nosso lado, ao pé, ao pôr-do-sol, ao redor, ao revés, aos poucos, ao seu lado, ao singelo, ao teu lado, ao todo, ao viés, ao vivo, a ouro e fio, à parte, a pau, a paulada, a pauladas, a pé, a pé quedo, a pelo, a picareta, a pique, a pleno, a pontapés, a postos, a pouco e pouco, a pressa, à primeira vista, a princípio, a propósito, a própria, a recado, a rédea solta, a regalada, a reio, a revelia, a revezes, a rigor, a risca, a rodo, a sabendas, a sabor, à saída, às apalpadelas, às avemarias, às ave-marias, às avessas, às bandeiras despregadas, às cavaleiras, às cegas, às claras, às costas, às direitas, a seco, a sério, às escondidas, às escuras, às fincas, às furtadas, a meu talante, a seu talante, as mais das vezes, as mais vezes, às mancheias, às mãos ambas, às mãos lavadas, a socapa, a solapa, a sorrelfa, a sós, às pauladas, às pressas, às rebatinhas, às singelas, assim assim, assim como, assim como assim, à sua disposição, à sua espera, à sua vista, à sua volta, à sua vontade, a súbitas, a surdina, às vezes, às tontas, à tarde, à tardinha, até certo ponto, a tempo, a tempo e a hora, a tempo e a horas, a tempo e hora, até pelo contrário, a tinta, à toa, à toda, a toda a hora, a toda hora, a todo o pano, a todo o pulso, à tona, a torto e a direito, a trecheio, a trecho, a tripa forra, a trouxe-mouxe, à tua custa, à tua disposição, à tua espera, à tua vista, à tua volta, à tua vontade, à uma, à uma hora, a unha de cavalo, a unhas de cavalo, a vau, à vela, a ventura, a vezes, à vista, a voga arrancada, à volta, à vontade, a vozes, bastante devagar, bem assim, bem longe da cidade, bem tarde, cada vez mais, cada vez menos, certas vezes, com amor, com calma, com certeza, com desconfiança, com desgosto, com exatidão, com gosto, com jeito, com medo, com muito jeito, com pressa, como tal, daqui a bocado, daqui a pouco, daqui a um bocadinho, daqui a um bocado, de acordo, de afogadilho, de alguma forma, de alto a baixo, de assento, de beijado, de bom grado, de cabo a rabo, de cada vez, de caso pensado, de cara, de certa maneira, de certo, de certo modo, de chapa, de chofre, de cima, de cima a baixo, de cima em baixo, de cor, de costas, de cotio, de dentro, de dia, de enviés, de esguelha, de espaço, de estudo, de fato, de fio a pavio, de fora, de fora parte, de forma alguma, de forma nenhuma, de frente, de golpe, de gosto, de graça, de improviso, de indústria, de jeito algum, de jeito nenhum, de joelhos, de lado, de largo, de leve, de longe, de longe a longe, de longe em longe, de má vontade, de maneira alguma, de maneira nenhuma, de manhã, de mansinho, de mão beijada, de mão em mão, de menos, de modo algum, de modo geral, de modo nenhum, de molde, de momento a momento, de muito, de nenhum modo, de noite, de norte a sul, de novo, dentro em breve, dentro em pouco, de oitiva, de onde em onde, de ouvida, de palanques, de parte a parte, de passagem, de perto, de pé, de pé atrás, depois de amanhã, de ponto em branco, de pouco, de preferência, de presente, de presto, de primeiro, de propósito, de qualquer jeito, de qualquer forma, de qualquer maneira, de qualquer modo, de quando em quando, de quando em vez, de raiz, de regra, de relance, de repelão, de repente, de resto, de revés, de rojo, de roldão, de rota batida, de salto, desde já, de segunda mão, de sobre, de sobreaviso, de sobrerrolda, de sobre-rolda, de sobressalto, de soslaio, de súbito, de tarde, de tempos a tempos, de tempos em tempos, de toda a parte, de toda forma, de toda maneira, de toda parte, de todo, de trás, de través, de tropel, de uma maneira geral, de uma vez, de um golpe, de um tiro, de verdade, de vereda, de vez, de vez em quando, de vez em vez, de vista, de viva voz, de volta, dia a dia, dia-a-dia, diversas vezes, do mesmo modo, do meu lado, do nosso lado, do seu lado, dos pés à cabeça, do teu lado, dum modo geral, duma maneira geral, duma vez, eis senão quando, em absoluto, em barda, em breve, em caixa, em caminho, em certa medida, em cheio, em cima, em conjunto, em conta, em contacto, em demasia, em especial, em excesso, em frente, em geral, em hipótese alguma, em mão, em média, em meu lugar, em nosso lugar, em parte, em particular, em pé, em ponto, em primeira mão, em princípio, em público, em que pé, em que pé que, em redor, em regra, em revés, em roda, em seguida, em segunda mão, em seu lugar, em série, em silêncio, em som de guerra, em suma, em surdina, em tempo, em termos, em teu lugar, em toda a parte, em toda a volta, em través, em vão, em verdade, em via de regra, em vista, em volta, entre a cruz e a caldeirinha, entre lusco e fusco, fora de mão, fora parte, frente a frente, graças a deus, grosso modo, gota a gota, hoje de noite, hoje em dia, inda agora, inda bem, inda por cima, inda assim, intúmeras vezes, lá dentro, lado a lado, lá em cima, lá embaixo, lá fora, mais cedo ou mais tarde, mais e mais, mais ou menos, menos mal, mercê de deus, muita vez, muitas das vezes, muitas vezes, muito cedo, muito rapidamente, muito tarde da noite, na certa, na medida do possível, na mesma, não raro, nas imediações, na verdade, nem mais nem menos, nem por isso, nem sempre, nem tanto, nenhuma vez, nesse caso, nesse comenos, nesse entremeio, nesse meio tempo, no conjunto, no fim, no fundo, no gênero, no geral, noite e dia, num átimo, nunca mais, nunca por nunca, o mais das vezes, outra vez, outras vezes, outro dia, outro tanto, p-a-pá santa justa, para a frente, para a semana, para baixo, para cima, para cima e para baixo, para dentro, para diante, para fora, para frente, para já, para lá, para menos, para o ano, para o lado, para o nosso lado, para o seu lado, para o teu lado, para onde, para todo o sempre, para trás, parte fora, passo a passo, pela frente, pela manhã, pela rama, pelo contrário, pelo meio, por acaso, por agora, por ali, por aqui, por artes de berlinques e berloques, por atacado, por baixo, por bem, por certo, por cima, por completo, por conta, por dá cá aquela palha, por dentro, por detrás, por enquanto, por fora, por força, por fim, por gosto, por hoje, por

Locuções Conjunctorias – LCONJ

a fim de que, agora que, ainda que, além de que, além disso, à medida que, a menos que, a não ser que, antes que, ao mesmo tempo, ao mesmo tempo que, ao passo que, apesar de que, à proporção que, assim como, assim que, até que, bem como, bem que, cada vez que, com tal que, como que, como quer que, como se, contanto que, dado que, daí que, da mesma maneira que, de cada vez que, de forma que, de jeito que, de maneira que, de modo que, de molde que, depois que, de que, desde a hora que, desde o momento que, desde que, de sorte que, dessa forma, de tal modo que, de tal sorte que, do mesmo modo que, do que, eis que, eis senão que, em vista disso, enquanto que, entretanto que, exceto se,inda que, já que, logo que, mas ainda, mas também, mesmo que, muito embora, nada obstante, na medida em que, não apenas, não obstante, não só, não somente, nem que, no caso que, no entanto, para que, pois que, por conseguinte, por consequência, por isso que, por mais que, por maior que, por melhor que, por menor que, por menos que, por muito que, por outro lado, por pior que, por pouco que, por que, por sua vez, posto que, primeiro que, quanto maior, quanto mais...mais, quanto mais...menos, quanto melhor, quanto menor, quanto menos...mais, quanto menos...menos, quanto pior, que nem, salvo se, se bem que, sem que, sempre que, senão ainda, suposto que, tal como, tal qual, tanto assim que, tanto como, tanto mais, tanto menos, tanto quanto, tanto que, tão logo que, todas as vezes que, uma vez que, visto como, visto que

Locuções Denotativas – LDEN

afinal de contas, além disso, além do que, a saber, com efeito, de mais a mais, diante disso, em conclusão, em resumo, em todo caso, em todo o caso, em virtude disso, em vista disso, espera aí, espera lá, mesmo assim, na verdade, não só, nesse ínterim, neste comenos, no fim das contas, ora bem, ou antes, ou seja, ou melhor, pelo contrário, pelo visto, pois bem, pois claro, por exemplo, por fim, por outro lado, por um lado, quer dizer, salvo erro

Locuções Prepositivas – LPREP

abaixo de, à borda de, à busca de, a caminho de, a cargo de, acerca de, acima de, a coberto de, à conta de, à custa de, a despeito de, adiante de, à direita de, à disposição de, à distância de, à espera de, à esquerda de, à exceção de, a favor de, a fim de, à flor de, à força de, à frente de, além de, a mandado de, a mando de, à maneira de, à margem de, à mercê de, a nível de, antes de, ao abrigo de, ao através de, ao cabo de, ao contrário de, ao encontro de, ao fim de, ao invés de, ao lado de, ao largo de, ao longo de, ao mandado de, ao mando de, ao modo de, ao nível de, ao par de, ao pé de, ao peso de, ao redor de, ao sabor de, aos cuidados de, ao termo de, a par de, a partir de, apesar de, a peso de, a poder de, a ponto de, à procura de, a propósito de, aquém de, a respeito de, à roda de, às custas de, às escondidas de, à sombra de, até a, à tona de, atrás de, através de, a ver com, à vista de, à volta com, à volta de, cerca de, com a intenção de, com base em, com o intuito de, com o propósito de, com relação a, com respeito a, com risco de, com vista a, com vistas a, da parte de, de acordo com, debaixo de, de conformidade com, de cima de, de dentro de, de encontro a, de fora de, de forma a, defronte de, de jeito a, de maneira a, de mistura com, de modo a, de molde a, dentro de, dentro em, de par com, depois de, de trás de, devido a, diante de, do gênero de, do lado de, em apoio a, em apoio de, em atenção a, embaixo de, em benefício de, em busca de, em caso de, em cima de, em comparação com, em conformidade com, em contraste com, em decorrência de, em face a, em face de, em favor de, em forma de, em frente a, em frente de, em função de, em lugar de, em meio a, em meio de, em oposição a, em prol de, em que pese a, em redor de, em relação a, em roda de, em termos de, em tomo a, em torno de, em troca de, em vez de, em via de, em vias de, em virtude de, em vista de, em volta de, face a, fora de, graças a, junto a, junto com, junto de, longe de, mercê de, na base de, na conformidade de, na conta de, não obstante, no alto de, no caso de, no centro de, no fim de, no fundo de, no gênero de, no interior de, no meio de, nos arredores de, no sentido de, para além de, para baixo de, para cima de, para cá de, para cima de, para com, para debaixo de, para dentro de, para fora de, para lá de, para o lado de, para trás de, pelo meio de, perto de, por baixo de, por causa de, por cima de, por conselho de, por conta de, por culpa de, por debaixo de, por dentro de, por detrás de, por diante de, por entre, por fora de, por força de, por mandado de, por mando de, por meio de, por menos de, por motivo de, por ocasião de, por trás de, por via de, por volta de, por vontade de, quanto a, relativamente a, respeito a, sem embargo de, sob pena de, um bocadinho de

Locuções Pronominais – LP

a gente, alguma coisa, aquele outro, cada qual, cada um, cada uma, comigo mesma, comigo mesmo, comigo própria, comigo próprio, conosco mesmas, conosco mesmos, consigo mesma, consigo mesmas, consigo mesmos, contigo mesma, contigo mesmo, ela mesma, elas mesmas, ela própria, elas próprias, ele mesmo, eles mesmos, ele próprio, eles próprios, eu mesma, eu mesmo, eu própria, eu próprio, fosse o que fosse, fossem quais fossem, fosse qual fosse, fosse quem fosse, mim mesma, mim mesmo, mim própria, mim próprio, nós mesmas, nós mesmos, nós próprias, nós próprios, o mais, o meu, o qual, o que, o que quer que, o que quer que seja, os quais, onde quer que, quaisquer que sejam, qualquer um, qualquer que fosse, qualquer que seja, quando quer que, quanto quer que, quem quer, quem quer que, quem quer que seja, seja de que maneira for, seja onde for, seja o que for, seja qual for, seja quem for, sejam quais forem, si mesma, si mesmas, si mesmo, si mesmos, si própria, si próprias, si próprio, si próprios, tal e qual, tal e tal, ti mesma, ti mesmo, ti própria, ti próprio, todo aquele, todo aquele que, tu mesma, tu mesmo, tu própria, tu próprio, um ou outro, você mesma, você mesmo, vocês mesmas, vocês mesmos, você própria, vocês próprias, você próprio, vocês próprios

Residual – RES

Todas as palavras que não tiverem sido etiquetadas até agora serão etiquetadas de acordo com a etiqueta que estiver associada a ela no léxico. Caso a palavra não conste no léxico serão utilizados os léxicos de sufixos e o de prefixos, na tentativa de formar uma palavra que faça parte do léxico. Se ainda assim a palavra for uma palavra desconhecida – em caso de iniciar com letra minúscula será etiquetada como RES e em caso de iniciar com letra maiúscula será etiquetada como NP.

Lista de regras contextuais

Mude a etiqueta de X para Y	Se a palavra atual é W	E a condição Z acontece
ART – PPOA	O, a, os, as	Se a etiqueta+1/etiqueta-1 é V*
ART – PD	O, a, os, as	Se a palavra+1 é que
ART – N	O, a, os, as, um	Se a etiqueta-1 for ART ou PPS
ART – PREP	A	Se a etiqueta-1 for VTI
ART – PIND	Um, uma, uns, umas	Se a palavra+1 ou palavra+2 é que. Ou se etiqueta +1 é ... ou . ou ! ou ?.
	Um, uma, uns, umas	Ou se a etiqueta+1 é V*.
	Um, uma	Ou se a palavra +1 é qualquer.
	Uns, umas	Ou se a palavra +1 é quaisquer.
ART – NC	Um, uma	Se a palavra+1 é e ou a palavra+1 é mais ou a palavra+1 é só ou a palavra+1 é menos ou a palavra+1 é vezes. Ou se palavra+1 é dividido e a palavra+2 é por.
PPOA – PAPASS	Se	Se a etiqueta+1 ou a etiqueta-1 é VTD ou VBI
PPOA – CONJSUB	Se	Se a etiqueta-1 era ... ou . ou ! ou ? ou , (início de período).
PINT – PR	Quem	Se a etiqueta-1 é pronome pessoal ou PREP ou V*
	Qual	Se a etiqueta-1 é ART
	Quando	Se a etiqueta+1 é N ou V*
	Quanto	Se a palavra-1 é tudo ou toda ou toda ou todos ou todas.
PINT – PIND	Onde	Se a etiqueta-1 é N ou PREP
	Qual	Se a etiqueta+1 é VAUX ou VTD ou VTI ou VBI ou VINT e a frase termina com ?.
	Quanto	Se a palavra+1 é mais ou se a palavra+1 é menos. Ou se etiqueta-1 é V* e a etiqueta+1 é V*.
PINT – CONJSUB	Qual	Se a etiqueta+1 é N e a etiqueta-1 é , e a etiqueta+2 ou etiqueta+3 é , (faz parte de um aposto)
	Quando	Se a etiqueta-1 era ... ou . ou ! ou ? (início de período) ou , (início de oração subordinada)
PINT – ADJ	Qual	Se a etiqueta+1 é N e o última palavra do período <u>não é uma ?</u> (frases declarativas)
PINT – ADV	Quanto	Quando o período termina com ! (períodos exclamativas).
PR – PINT	Que	Ou se a palavra+1 é custa
PR – PIND	Que	Se é a primeira palavra da frase e a frase termina com ?
PR – CONCOORD	Que	Se a etiqueta+1 é N e a frase termina com . ou !
PR – CONJSUB	Que	Se a palavra-1 é uma das conjugações dos verbos dizer, falar, afirmar, declarar, jurar
PPS – ADJ	Meu , minha , meus , minhas , teu , tua , teus , tuas , seu , sua , seus ,	Se a etiqueta-1 é , Se a etiqueta+1 é N

PD – ADJ	suas , nosso , nossa , nossos , nossas , vosso , vossa , vossos , vossas , seu , sua , seus , suas este, esta , estes , estas , isto , esse , essa , esses , essas , isso , aquele , aquela , aqueles , aquelas , aquilo	Se a etiqueta+1 é N
PIND – ADV	Algo, todo, toda Pouco, demais, mais Mais	Se a etiqueta+1 é ADJ Se a etiqueta-1 é V* Nas expressões cada...mais, mais+adj+que, não mais, a mais, de mais a mais, mais dia, menos dia, mais pra lá do que pra cá Se a etiqueta-1 é NC e a etiqueta+1 é NC Se a etiqueta+1 ou etiqueta-1 é N
PIND – ADJ	Mais Algum, nenhum, qualquer, tal, vários, várias Pouco, outro, cada, Todo, toda Vário, varia	Se a etiqueta+1 é N Se a etiqueta+1 é ART Se a etiqueta-1 é N
PIND – N	Todo Pouco, mais	Se a etiqueta-1 é ART ou PREP+ART
PREP – ADV	Afora, após Após Até Até Menos	Se a etiqueta-1 é ART Se a etiqueta+1 for . ou , Se a etiqueta+1 não é ART ou N Se a etiqueta+1 é PREP Se a etiqueta+1 é Verbo no gerúndio Se a etiqueta-1 é V*
PREP – ADJ	consoante	Se a etiqueta-1 <i>verbo ser ou estar</i> ou se a etiqueta-1 é N
PREP – CONJSUB	Mediante, salvo Menos, salvo Salvo conforme	Se a etiqueta-1 N Se a etiqueta+1 N Se a etiqueta-1 é VLIG Se a etiqueta+1 ou etiqueta+2 ou etiqueta+3 é V*
PREP – N	consoante Consoante, contra, durante, exceto, fora, malgrado, mediante, menos	Se a etiqueta-1 é , Se a etiqueta-1 é ART
PREP – PDEN	contra menos	Se a etiqueta-1 é verbo ser Se a etiqueta-1 PREP+ART
PREP – VINT	sobre	Se a etiqueta-1 é , e a etiqueta+1 é pronome ou N e a etiqueta+2 é . ou ,
PREP – VTI	sobre	Se a palavra-1 é que e a próxima não é prep (se a palavra-2 é que e a palavra-1 é pronome pessoal) ou se a palavra-1 é que e a etiqueta+1 é PREP
ADverbo – ADJ	muito	Se a etiqueta+1 é N
ADverbo – N	muito	Se a etiqueta-1 é ART
ORD – PREP	segundo	Se a etiqueta+1 é N ou NP ou ART
ORD – N	segundo	Se a etiqueta+1 e a etiqueta-1 não são N
CONCOORD – N	E, logo, mas, porém	Se a etiqueta-1 é ART
CONCOORD – ADV	Já	Se a etiqueta-1 ou a etiqueta+1 é V*
CONCOORD – PDEN	Logo	Se a etiqueta-1 é V*
	Mas	Se a etiqueta-1 é VAUX ou VTD ou VTI ou VBI ou VINT e a etiqueta+1 é VLIG
verbo – CONCOORD	quer	Se a etiqueta+1 é verbo no subjuntivo
	seja	Se a palavra-1 e palavra-2 e palavra-3 não são que
verbo – CONJSUB	caso	Se é a primeira palavra do período ou se a etiqueta-1 é , e a palavra+1 <u>não</u> é com e a etiqueta+1 não é ADV

verbo – N CONJSUB – N CONJSUB – V	caso mal como	Se a etiqueta-1 é ART Se a etiqueta-1 é ART Se a palavra-1 é eu. Ou se não nenhuma outra palavra do período está etiquetada como verbo.
CONJSUB – ADV	como como	Se o período termina com ? ou ! Se a palavra-1 é uma das conjugações de: Conte, diga, descreva, fale (interrogação indireta)
CONJSUB – PREP CONJSUB – CONJCOORD	embora Pois	Se a etiqueta+1 é ART e a etiqueta+2 é N Se a etiqueta-1 é , e a etiqueta+1 é , . Ou se a etiqueta-1 é ;
N – ADJ	Qualquer que seja a palavra	Se a palavra-1 é muito ou tão ou bem. Se a palavra-1 é mais e a palavra a ser etiquetada estiver no singular. Se a etiqueta+1 ou se a etiqueta-1 é N. Se as etiqueta+1 é VLIG e a etiqueta+2 é ART. (Doce é a pera) Se a etiqueta-1 é VTD+PPOA ou PPOA ou VTD ou VTI + PPOA.
N – VTD	Palavras que apresentam os sufixos de verbo ²⁹ - a, ado, ais, am, amos, ando, ar, ara, aram, áramos, aras, ardes, ardes, áreis, arem, ares, armos, as, asse, ásseis, assem, ássemos, asses, aste, astes, ava, avam, ávamos, avas, áveis, e, ei, em, emos, endo, er, era, eram, éramos, eras, erdes, éreis, erem, eres, ermos, es, esse, ésseis, essem, éssemos, esses, este, estes, i, ia, iam, famos, ias, ido, ieis, ieis, imos, imos, indo, ir, ira, iram, iramos, iras, irdes, íreis, irem, ires, irmos, is, isse, ísseis, issem, íssemos, isses, iste, istes, iu, o, ou, rá, ram, rão, rás, rei, reis, remos, ria, riam, riamos, rias, rieis	Se a etiqueta-1 ou a etiqueta+1 é PPOA.
	*	Se a etiqueta+1 é ART e a etiqueta+2 é N.
	Palavras que apresentam os sufixos de verbo: <i>ado, ido</i>	Se a etiqueta-1 é VAUX (voz passiva)
N – VLIG	*	Se a etiqueta+1 é ADJ
	*	Se a etiqueta+1 é PREP
N – VAUX	*	Se a etiqueta+1 é V e a palavra+1 tem os sufixos: <i>ando, endo, indo, ado, ido</i>
N – VTI	*	Se a etiqueta+1 ou etiqueta+2 é PREP ou PREP+ART
N – VINT	*	Se a etiqueta+1 é ADV
N – VBI		verbos <i>dar, devolver, entregar, mostrar, oferecer, pedir, enviar</i>
	*	se o verbo for seguido de PPOA e a etiqueta+1 é PREP (absolvê-lo de...)
	*	Se a etiqueta+1 é N ou PREP ou PREP+ART (acertou Pedro no braço)

²⁹ O símbolo * será utilizado a partir deste ponto para indicar as palavras que terminem com este mesmo sufixo

APÊNDICE D — INCORPORA

O InCorpora é um software que integra 27 ferramentas — implementadas em ANSI C responsáveis por preparar *corpus* e textos para treinamento e etiquetagem, gerar estatísticas e listas de dados que auxiliem na avaliação dos métodos de etiquetagem, combinar as saídas dos etiquetadores utilizando os métodos descritos no Capítulo 5 e avaliar os resultados da combinação. O módulo *incorpora* não possui ainda uma interface gráfica fazendo com que o acesso às ferramentas se dê através de linhas de comando. Cada uma das ferramentas é detalhada abaixo.

D1 CorpusTT

Entrada: corpus etiquetado no formato palavra_etiqueta com um período por linha

Saída: corpus etiquetado no formato palavra TAB etiqueta com uma palavra por linha

Finalidade: colocar um corpus de treinamento no formato pedido pelo etiquetador TreeTagger

Linha de comando: CorpusTT <corpus1> <corpus2>

Onde:

TAB é tabulação

corpus1 é o arquivo de entrada

corpus2 é o arquivo de saída

D2 CorpusTBL

Entrada: corpus etiquetado no formato palavra_etiqueta com um período por linha

Saída: corpus etiquetado no formato palavra/etiqueta com um período por linha

Finalidade: colocar um corpus de treinamento no formato pedido pelo etiquetador TBL

Linha de comando: CorpusTBL <corpus1> <corpus2>

Onde:

corpus1 é o arquivo de entrada

corpus2 é o arquivo de saída

D3 Lexify

Entrada: corpus etiquetado no formato palavra_etiqueta com um período por linha

Saída: corpus etiquetado no formato palavra@etiqueta/outraet₁.../outraet_n com um período por linha finalizado por //

Finalidade: colocar um corpus de treinamento no formato pedido pelo etiquetador neural elástico

Linha de comando: Lexify <*corpus1*> <*corpus2*>

Onde:

Outraet_i com i= 1...n são as n outras etiquetas possíveis para aquela palavra segundo um léxico

corpus1 é o arquivo de entrada

corpus2 é o arquivo de saída

D4 Lexicon

Entrada: corpus etiquetado no formato palavra_etiqueta com um período por linha

Saída: léxico no formato palavra_outraet₁-outraet₂-...-outraet_n com uma palavra por linha

Finalidade: gerar um léxico a partir do corpus de treinamento para ser utilizado pela ferramenta Lexify e pelo PoSiTagger

Linha de comando: Lexicon <*corpus1*> <*léxico*>

Onde:

outraet_i com i= 1...n são as n etiquetas possíveis para uma dada palavra

corpus1 é o arquivo de entrada

léxico é o arquivo de saída

D5 TTLexicon

Entrada: corpus etiquetado no formato palavra_etiqueta com um período por linha

Saída: léxico no formato palavraTABoutraet₁TAB-TABoutraet₂TAB-...TABoutraet_n

Finalidade: gerar um léxico a partir do corpus de treinamento para ser utilizado no treinamento do etiquetador TreeTagger

Linha de comando: TTLexicon <*corpus1*> <*léxico*>

Onde:

TAB é tabulação

outraet_i com i= 1...n são as n etiquetas possíveis para uma dada palavra

corpus1 é o arquivo de entrada

léxico é o arquivo de saída

D6 TokenizerTT

Entrada: texto não etiquetado

Saída: texto não etiquetado com uma palavra/símbolo por linha

Finalidade: colocar um texto a ser etiquetado no formato exigido pelo etiquetador TreeTagger

Linha de comando: TokenizerTT <*texto1*> <*texto2*>

Onde:

texto1 é o arquivo de entrada

texto2 é o arquivo de saída

D7 Tokenizer

Entrada: texto não etiquetado

Saída: texto não etiquetado com as palavras separadas dos símbolos por espaço em branco, no formato um período por linha

Finalidade: colocar um texto a ser etiquetado no formato exigido pelo etiquetador TBL e MXPOST

Linha de comando: Tokenizer <texto1> <texto2>

Onde:

texto1 é o arquivo de entrada

texto2 é o arquivo de saída

D8 Ambiguous

Entrada: texto não etiquetado no formato um período por linha

Saída: apresenta na tela o número de palavras ambíguas

Finalidade: calcular o número de palavras ambíguas presentes em um texto

Linha de comando: ambiguous <léxico> <texto>

Onde:

texto é o texto para o qual deve ser calculado o número de palavras ambíguas

léxico é o léxico que servirá de base para que o programa decida se uma palavra é ou não ambígua

D9 Diff

Entrada:

- 1) corpus de treinamento etiquetado no formato palavra_etiqueta com um período por linha
- 2) texto etiquetado no formato palavra_etiqueta com um período por linha

Saída: arquivo com o número de palavras desconhecidas e a lista destas palavras

Finalidade: verificar o número de palavras desconhecidas presentes em um texto

Linha de comando: diff <corpustrain> <texto> <arquivosaída>

Onde:

corpustrain é o corpus de treinamento que será tomado como base para saber se uma palavra é desconhecida, ou por não ter aparecido no treinamento ou por não ter aparecido no treinamento com uma determinada etiqueta

texto é o texto para o qual deve ser calculado o número de desconhecidas

arquivosaída é o arquivo com o número de palavras desconhecidas e uma lista de quais são elas

D10 Comp

Entrada:

- 1) texto etiquetado pelo TreeTagger, ou TBL, ou MXPOST, ou PoSiTagger, ou Neural Elástico
- 2) texto manualmente etiquetado no formato palavra_etiqueta

Saída: arquivo com a avaliação do etiquetador contendo: número de palavras, número de erros, precisão geral, precisão por etiqueta, precisão por grupo

Finalidade: avaliar os resultados dos etiquetadores

Linha de comando: Comp <texto1> <texto2> <avaliação>

Onde:

texto1 é o texto etiquetado automaticamente

texto2 é o texto etiquetado manualmente

avaliação é o arquivo de saída que contém o número de palavras do texto, o número de erros, a precisão geral e as precisões por etiquetas

D11 Lista

Entrada: arquivo com a lista dos arquivos de configuração de cada um dos etiquetadores que serão comparados

Saída: arquivo com a lista dos erros cometidos da mesma forma por todos os etiquetadores

Finalidade: gerar uma lista com os erros cometidos da mesma forma por todos os etiquetadores com a etiqueta correta para aquele caso

Linha de comando: lista <*listacconfig*> <*listaeerros*>

Onde:

listacconfig é o arquivo de entrada com a lista dos arquivos de configuração dos etiquetadores no formato:

etiquetador1.txt
etiquetador2.txt

...
etiquetadorn.txt

Onde:

etiquetadorn.txt é o arquivo de configuração do etiquetador n no formato:

@tagger configuration@v1.0@
[NomeEtiquetador]
log_file=arquivolog
tagged_file=arquivoetiquetador
train_file=arquivomanual
tagset_file=conjuntoetiquetas

Onde:

arquivolog é o arquivo gerado por este programa que contém os dados da avaliação do etiquetador — precisão geral e precisão por etiquetas

arquivoetiquetador é o arquivo etiquetado pelo etiquetador no formato palavra_etiqueta

arquivomanual é o arquivo etiquetado manualmente no formato palavra_etiqueta

conjuntoetiquetas é o conjunto de etiquetas.

listaeerros é o arquivo de saída no formato: palavra etiqueta1 etiqueta2 ... etiquetan etiquetacorreta. Onde etiqueta1 é a etiqueta de saída do etiquetador1, etiqueta2 é a etiqueta de saída do etiquetador2, etiquetan é a etiqueta de saída do etiquetadorn e etiquetacorreta é a etiqueta do texto manualmente etiquetado.

D12 Lista2

Entrada: arquivo com a lista dos arquivos de configuração de cada um dos etiquetadores que serão comparados

Saída: arquivo com a lista dos casos em que todos os etiquetadores erraram

Finalidade: gerar uma lista com os em que todos os etiquetadores erraram

Linha de comando: lista2 <*listacconfig*> <*listaeerros*>³⁰

D13 Bootstrapping

Entrada: Corpus manualmente etiquetado no formato palavra_etiqueta

³⁰ Para a ferramenta Lista2 os arquivos listacconfig e listaeerros são como os da ferramenta Lista1

Saída: n pares de corpus de treinamento (.bs) e teste (.bst) no formato palavra_etiqueta

Finalidade: gerar uma aproximação da precisão geral verdadeira utilizando o método *e0 bootstrapping* descrito na Seção 4.1.3.1

Linha de comando: bootstrap <corpus> <n>

Onde:

corpus é o corpus a partir do qual o programa irá gerar n corpus de treinamento e teste

n é o número de duplas corpus de treinamento/corpus de teste que deverá ser gerado

D14 Combin

Entrada: arquivos etiquetados por dois etiquetadores

Saída: a taxa de complementaridade dos etiquetadores

Finalidade: verificar qual é a taxa de complementaridade entre dois etiquetadores

Linha de comando: combin <texto1> <texto2>

Onde:

Texto1: é o texto etiquetado pelo etiquetador 1

Texto2: é o texto etiquetado pelo etiquetador 2

D15 Random

Entrada: arquivos etiquetados pelos etiquetadores que serão combinados

Saída:

1) arquivo com o resultado da combinação no formato palavra_etiqueta

2) arquivo com a avaliação da combinação com o número de palavras, número de palavras que todos os etiquetadores etiquetaram corretamente, número de palavras que todos os etiquetadores etiquetaram errado, número de vezes em que a maioria dos etiquetadores etiqueta corretamente, número de vezes em que a minoria dos etiquetadores etiqueta corretamente, número de vezes que pelo menos um dos etiquetadores etiqueta corretamente, precisão geral obtida pelo método de combinação, precisão geral que poderia ter sido alcançada, precisão geral de cada etiquetador individual, precisão por etiquetas do método e dos etiquetadores.

Finalidade: combinar as saídas de vários etiquetadores aleatoriamente e avaliar os resultados desta combinação

Linha de comando: random <listataggers> <textocombinado> <avalcombinacao>

Onde:

listataggesr lista com os arquivos de configuração dos etiquetadores no formato:

etiquetador1.txt

etiquetador2.txt

...

etiquetadorn.txt

Onde:

etiquetadorn.txt é o arquivo de configuração do etiquetador n no formato:

@tagger configuration@v1.0@

[NomeEtiquetador]

log_file=arquivolog

tagged_file=arquivoetiquetador

train_file=arquivomanual

tagset_file=conjuntoetiquetas

Onde:

arquivolog é o arquivo gerado por este programa que contém os dados da avaliação do etiquetador — precisão geral e precisão por etiquetas

arquivoetiquetador é o arquivo etiquetado pelo etiquetador no formato palavra_etiqueta

arquivomanual é o arquivo etiquetado manualmente no formato palavra_etiqueta

conjuntoetiquetas é o conjunto de etiquetas.

textocombinado é o arquivo com o texto etiquetado através da combinação aleatória das saídas dos etiquetadores

avalcombinação é o arquivo que contém a avaliação do método de combinação

D16 Majority³¹

Finalidade: combinar as saídas de vários etiquetadores utilizando o método Majority1 descrito no Capítulo 5

Linha de comando: majority <listataggers> <textocombinado> <avalcombinacao>

D17 Simplev

Finalidade: combinar as saídas de vários etiquetadores utilizando o método Majority2 descrito no Capítulo 5

Linha de comando: simplev <listataggers> <textocombinado> <avalcombinacao>

D18 Major

Finalidade: combinar as saídas de vários etiquetadores utilizando o método Majority3 descrito no Capítulo 5

Linha de comando: major <listataggers> <textocombinado> <avalcombinacao>

D19 Totprec

Entrada: arquivos etiquetados pelos etiquetadores que serão combinados e corpus de calibração

Saída:

1) arquivo com o resultado da combinação no formato palavra_etiqueta

2) arquivo com a avaliação da combinação com o número de palavras, número de palavras que todos os etiquetadores etiquetaram corretamente, número de palavras que todos os etiquetadores etiquetaram errado, número de vezes em que a maioria dos etiquetadores etiqueta corretamente, número de vezes em que a minoria dos etiquetadores etiqueta corretamente, número de vezes que pelo menos um dos etiquetadores etiqueta corretamente, precisão geral obtida pelo método de combinação, precisão geral que poderia ter sido alcançada, precisão geral de cada etiquetador individual, precisão por etiquetas do método e dos etiquetadores.

Finalidade: combinar as saídas de vários etiquetadores utilizando o método TotPrecision descrito no Capítulo 5

Linha de comando: totprec <listataggers> <textocombinado> <avalcombinacao>

Onde:

³¹ As definições de entrada, saída, listatagger, textocombinada e avalcombinação são as mesmas da ferramenta D15 e também valem para as ferramentas D17 e D18.

listataggers lista com os arquivos de configuração dos etiquetadores no formato:

etiquetador1.txt

etiquetador2.txt

...

etiquetadorn.txt

Onde:

etiquetadorn.txt é o arquivo de configuração do etiquetador n no formato:

@tagger configuration@v1.0@

[NomeEtiquetador]

log_file=arquivolog

tagged_file=arquivoetiquetador

train_file=arquivomanual

tune_man_file=calibraman

tune_train_file=calbraet

tagset_file=conjuntoetiquetas

[

Onde:

arquivolog é o arquivo gerado por este programa que contém os dados da avaliação do etiquetador — precisão geral e precisão por etiquetas

arquivoetiquetador é o arquivo etiquetado pelo etiquetador no formato palavra_etiqueta

arquivomanual é o arquivo etiquetado manualmente no formato palavra_etiqueta

calibraman é o corpus de calibração no formato palavra_etiqueta

calbraet é o corpus de calibração etiquetado pelo etiquetador no formato palavra_etiqueta

conjuntoetiquetas é o conjunto de etiquetas.

textocombinado é o arquivo com o texto etiquetado através da combinação aleatória das saídas dos etiquetadores

avalcombinação é o arquivo que contém a avaliação do método de combinação

D20 Tagprec³²

Finalidade: combinar as saídas de vários etiquetadores utilizando o método TagPrecision descrito no Capítulo 5

Linha de comando: tagprec <*listataggers*> <*textocombinado*> <*avalcombinacao*>

D21 Precall

Finalidade: combinar as saídas de vários etiquetadores utilizando o método Precision-Recall descrito no Capítulo 5

Linha de comando: precall <*listataggers*> <*textocombinado*> <*avalcombinacao*>

D22 Pairwise

Finalidade: combinar as saídas de vários etiquetadores utilizando o método TagPair descrito no Capítulo 5

Linha de comando: pairwise <*listataggers*> <*textocombinado*> <*avalcombinacao*>

D23 Stack_pair

Finalidade: combinar as saídas de vários etiquetadores utilizando o método Tags descrito no Capítulo 5

Linha de comando: stack_pair <listataggers> <textocombinado> <avalcombinacao>

D24 Stack_pair_word

Finalidade: combinar as saídas de vários etiquetadores utilizando o método Tags+word descrito no Capítulo 5

Linha de comando: stack_pair_word <listataggers> <textocombinado> <avalcombinacao>

D25 Picktag

Finalidade: combinar as saídas de vários etiquetadores utilizando o método Pick tag descrito no Capítulo 5

Linha de comando: picktag <listataggers> <textocombinado> <avalcombinacao>

D26 Picktagger

Finalidade: combinar as saídas de vários etiquetadores utilizando o método Pick tagger descrito no Capítulo 5

Linha de comando: picktagger <listataggers> <textocombinado> <avalcombinacao>

D27 Bagging

Entrada: Corpus de treinamento manualmente etiquetado no formato palavra_etiqueta

Saída: n corpus de treinamento

Finalidade: gerar aleatoriamente n corpus de treinamento a partir do corpus original e com o mesmo tamanho do corpus original para serem utilizados no método de combinação bagging descrito no capítulo 5

Linha de comando: bagging <corpustrein> <n>

Onde:

corpustrein é o corpus de treinamento a partir do qual o programa irá gerar os n corpus de treinamento

n é o número de corpus de treinamento que deverá ser gerado

³² As definições de entrada, saída, listatagger, textocombinado e avalcombinação são as mesmas da ferramenta D19 e também valem para as ferramentas D21, D22, D23, D24, D25 e D26.

GLOSSÁRIO

Algoritmo back-propagation

É um algoritmo supervisionado, em que as instâncias de treinamento (entrada, saída desejada) são utilizadas por um mecanismo de correção de erros cuja peculiaridade é o aprendizado dos pesos da rede em duas fases — chamadas de fase forward e fase backward. A fase forward é utilizada para definir a saída para um dado padrão de entrada (Braga et al., 1998). Dada uma rede com n camadas, o algoritmo seria como o mostrado abaixo.

Repete

Para cada instância de treinamento propague as entradas pela rede } Forward
 { Se a saída da rede não estiver próxima o suficiente da saída desejada
 então
 Para cada camada da rede, começando na camada n até a camada 1
 Para cada nó nesta camada
 Backward Ajuste os pesos do nó
 Até que a saída esteja próxima o suficiente da saída desejada para cada instância

Algoritmo da Freqüência Relativa

Faz a estimativa dos parâmetros de um HMM, baseando-se nas freqüências dos padrões que ocorrem em um dado corpus etiquetado. Calcula-se a freqüência com que uma palavra p_i ocorre com a etiqueta tag_i (freqüência lexical) e o número de vezes de que uma etiqueta tag_i é precedida pelas etiquetas tag_{i-2} e tag_{i-1} (freqüência contextual).

Algoritmo de Viterbi

É uma maneira simples e otimizada de fazer o cálculo do caminho mais provável (sequência de estados percorridos para gerar um período). Analisa um símbolo da sequência a cada espaço de tempo, começando pelo símbolo inicial. Para o símbolo que está sendo analisado o algoritmo calcula o caminho mais provável para chegar a cada um dos estados existentes ou seja, é feito o cálculo do melhor caminho resultante em cada um dos estados (Viterbi, 1967).

Algoritmo Forward-Backward

É utilizado para fazer o treinamento de HMMs. Utiliza probabilidades forward e backward para ajustar os parâmetros do HMM, de modo que seja atribuída a maior probabilidade possível a sequência de treinamento analisada. Calcula recursivamente dois conjuntos de probabilidades: a forward e a backward. A probabilidade forward é a probabilidade de em um dado instante de tempo (t), dada uma sequência de símbolos $\{p_1, p_2, \dots, p_{t-1}\}$ o HMM estar em um estado S_t . Já a probabilidade backward é a probabilidade de estando em um estado S_t , no tempo t , se ter a sequência $\{p_t, \dots, p_n\}$.

Classes abertas

Todas as classes de palavras que podem produzir palavras novas. Por exemplo: substantivo, verbo e adjetivo.

Classes de ambigüidade

Classes de ambiguidade são pares, triplas,...,n-ênuplas formados pelas etiquetas que possam ser associadas às palavras com mesma forma. Por exemplo, a palavra "casa", pode ser a palavra "casa" verbo ou a palavra "casa" substantivo, o mesmo vale para a palavra "mata". A classe de ambigüidade da qual farão parte estas palavras é a classe SUBSTANTIVO-VERBO, da qual também fazem parte outras palavras que têm sua ambigüidade na decisão de se são substantivo, ou verbo.

Classificadores

São ferramentas de aprendizado de máquina que classificam exemplos conhecidos e novos, utilizando-se para isto de um modelo que é gerado da extração de conhecimento de um conjunto de dados – escolha e adaptação de parâmetros da representação do modelo, através do paradigma escolhido (tais como simbólico, estatístico, e outros). O etiquetador morfossintático é um classificador que gera um modelo a partir de um corpus de treinamento (conjunto de dados) para etiquetar novos textos (novos exemplos) que podem conter palavras e contextos conhecidos e/ou desconhecidos.

Corpora

É o plural de corpus.

Corpus

A princípio qualquer coleção de mais de um texto pode ser chamada de corpus. Mas quando este termo é utilizado em lingüística moderna tende a ser utilizado com mais freqüência como uma coleção (conjunto) de textos representativos da língua, que tem tamanho finito, está disponibilizada em formato eletrônico e é um padrão de referência.

Constraint Grammar

Uma coleção de regras de ação-padrão, sendo apenas uma regra para cada forma ambígua de etiqueta. Cada regra especifica um ou mais padrões de contexto ou “restrições”, em que a etiqueta não é válida. Se algum destes contextos for satisfeita durante a desambigüização, a etiqueta é apagada. Estes padrões de contexto podem ser locais ou globais, e podem se referir a análises ambíguas ou não.

Dicionário Aberto

É constituído por todas as palavras do corpus de treinamento e por suas respectivas classes de ambigüidade.

Dicionário Fechado

É constituído por todas as palavras do corpus de treinamento e de teste e por suas respectivas classes de ambigüidade.

Hidden Markov Model (HMM)

É uma máquina de estado finito que é uma generalização da cadeia de Markov, em que assume-se que as transições de estado não foram observadas. No entanto, funciona como um gerador de sequências de observação aleatórias, conhecendo assim quais são as possíveis sequências.

Information Gain

Considera cada atributo isoladamente avaliando o quanto à informação fornecida por este atributo contribui para se saber qual é a classificação correta — ganho de informação. O ganho de informação que um dado atributo pode gerar é dado pela diferença de incerteza (por exemplo, entropia) entre situações sem e com conhecimento do valor deste atributo.

Máxima Entropia

Entropia é uma medida matemática de ignorância, é inversamente proporcional à informação, isto é, quando o número de informações diminui a entropia aumenta (alta entropia é sinônimo de alta ignorância). O conceito de entropia começou a ser utilizada no século 19 por J. Willard Gibbs para inferir propriedades termodinâmicas de sistemas físicos de seus valores de energia.

De acordo com a segunda lei da termodinâmica, entropia em um sistema fechado tende ao máximo (total desordem), de forma que a informação se dissipar mas não é alcançada. Uma situação bem estruturada e organizada que tenha baixa entropia precisa de menos informação para ser descrita, mas pode ser também resultado da adição de mais informação ao sistema. Máxima entropia pressupõe que todas as probabilidades são iguais e independentes umas das outras. Mínima entropia existe quando se esperar encontrar uma única possibilidade.

Modelo de Markov

É um processo estocástico com instantes discretos em que dada uma sequência aleatória de

variáveis $X = (X_1, \dots, X_T)$ que assumem valores de algum conjunto finito $S = \{s_1, \dots, s_N\}$, vale a seguinte propriedade:

$$\begin{aligned} 1) P(X_{t+1} = s_k | X_1, \dots, X_t) &= P(X_{t+1} = s_k | X_t) \\ &= P(X_2 = s_k | X_1) \end{aligned}$$

Chama-se X de cadeia de Markov ou diz-se que X tem as propriedades de Markov.

Ou seja, neste modelo uma variável depende somente da variável anterior e influencia somente a variável seguinte.

Modelo de n-gramas

Torna possível analisar a "vizinhança" de uma palavra, o contexto. Define que, para cada palavra, deverão ser analisadas as ' $n-1$ ' vizinhas. Os mais utilizados são o bigrama (Church & Gale, 1991) (é analisada apenas a palavra anterior) e trigrama (são analisadas as duas palavras anteriores).

Período

O termo sentence do inglês refere-se a período apesar de costumar ser traduzido equivocadamente como sentença. Um período é composto por um ou mais orações e/ou frases e tem como características:

- apresentação de um sentido ou significado completo;
- encerrar-se por meio de certos símbolos de pontuação.

Redes MLP (Multilayer Perceptron)

Redes MLP são redes perceptron multi-camadas utilizadas para resolver problemas não linearmente separáveis (Braga et al., 1998).