PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DESAMBIGUAÇÃO ESTOCASTICAMENTE EFICIENTE DE ANOTAÇÕES MORFOSINTÁTICAS FEITAS POR MTMDD

Área de Processamento de Linguagem Natural

por

Pablo Frederico Oliveira Thiele

Paulo Fernandes, Dr.
Orientador

Lucelene Lopes, Dra.
Co-Orientadora

Seminário de Andamento

Porto Alegre
2014

SUMÁRIO

LISTA DE ABREVIATURAS

CG	Constraint Grammar
HMM	Hidden Markov Models
PLN	Processamento de Linguagem Natural
POS	Part-of-Speech
POST	Part-of-Speech Tagging
SVM	Support Vector Machines
DTMC	Discrete Time Markov Chains
CTMC	Continuous Time Markov Chains

1 INTRODUÇÃO

Hodiernamente as tecnologias de Processamento de Linguagem Natural (PLN) estão sendo utilizadas em análises de enormes quantidades de dados. Com o advento das novas mídias e à adoção em massa das redes sociais, o fluxo de informações geradas a cada segundo é o maior da história. Segundo o SINTEF, 90% de todas as informações hoje armazenadas no planeta foram geradas nos últimos dois anos [1]. Embora isso se concentre, em maior parte, por informações e arquivos de multimídia, uma grande parcela da informação produzida, principalmente nas redes sociais, é textual. Desta forma, as soluções de PLN necessitam ser mais robustas do que jamais foram, encontrando soluções de processamento que possam acompanhar esta geração constante de informações ou pelo menos apresentar resultados melhores se comparados aos procedimentos utilizados anteriormente.

Tendo em mente a necessidade de velocidade de processamento necessário no caso de utilizarmos uma enorme quantidade de informação no procedimento, devemos também, manter a meta de obter a melhor taxa de acerto possível. Os etiquetadores ou *taggers* são um dos principais componentes da PLN, e como elemento crucial, antes de velocidade sua acuidade deve ser a premissa. Sua função, explorada nesse caso é a capacidade de observar e catalogar as palavras de um texto de acordo com suas funções morfossintáticas. O nome comumente dado a este processo é o de POST (Part-Of-Speech Tagging).

Dentro do contexto Part-Of-Speech (POS) encontra-se a função de processar e identificar um grupo de palavras agrupando-as em tipos pré-definidos. Este agrupamento pode ocorrer em razão sintática, morfológica ou morfossintática. Assim sendo, uma palavra pode ser definida como verbo, adjetivo ou pronome citando apenas algumas opções. Bem como, utilizando uma observação de contexto de sua frase, pode-se definir se uma determinada palavra atua como sujeito, objeto direto ou qualquer outra possibilidade adequada.

O conceito da obtenção de etiquetas semânticas a partir de avaliações dos textos embora pareça simples em um primeiro momento, apresenta vários desafios. Um dos maiores desafios encontrado em PLN é o problema da ambiguidade. Esta situação que ocorre nas mais diversas etapas do processamento de linguagem natural é complexa, devido à necessidade de que a aplicação processadora tenha conhecimentos abrangentes que possam ser utilizados como ferramentas que colaborem no intuito de realizar as escolhas mais corretas. Devido ao fato de se tratar de um

problema antigo, inerente à linguagem natural e existente desde o começo das pesquisas da área, diversas possibilidades de minimizar suas consequências foram propostas. O presente trabalho enumera algumas das propostas encontradas, adicionando a possibilidade de uso de estruturas do tipo MTMDD no processo, buscando um ganho substancial de desempenho.

1.1 Motivação

De acordo com Silva [2], a ambiguidade é uma das maiores dificuldades a serem devidamente administradas nos processos de linguagem natural. Suas peculiaridades recaem nas suas diversas formas de apresentação que podem ocorrer em cada etapa do processo. Suas variações podem ser do tipo que representa ambiguidade semântica, sendo esta mitigada com um conhecimento do mundo real do seu redor. Há também as ambiguidades de baixo nível, sendo estas exemplificadas como o reconhecimento correto do ponto final em uma frase, que pode ser tanto para marcar o final desta ou simplesmente indicar a presença de uma palavra que representa uma abreviatura.

Em um passo anterior à desambiguação, existe a necessidade das anotações POS das palavras de um texto de uma forma correta, essas etiquetas são obtidas após um pré-processamento do texto. O processo de etiquetagem que já foi em seus primórdios uma tarefa estritamente manual executada basicamente por especialistas do idioma do texto, hoje pode ser obtido com o processamento automatizado. As formas e objetivos dos atuais processamentos automáticos variam, alguns se utilizam de capacidades de aprendizagem de máquina para que após algumas fases de treino com grandes quantidades de texto, estes componentes que "aprendam" como os textos em um determinado domínio se comportam. Isso permite que esses programas possam ser mais efetivos quando a eles é apresentado um texto novo a fim de que se sejam anotadas todas as suas palavras.

Na literatura podem ser encontradas propostas de desambiguação que entremeiam os processos de etiquetação dos textos processados. Citando exemplos, temos os trabalhos Aduriz e Illarraza [3], Brill [4], Giménez e Màrquez [5] e Segond et.al.[7]. Basicamente cada proposta utiliza-se de um horizonte único, alguns trabalham com a noção de utilizar métodos estritamente probabilísticos no momento de definir as desambiguações adequadas. No entanto outros exemplos dão conta de utilizarem uma forma gramatical, criando regras específicas que são utilizadas como guias no momento da catalogação das palavras. Nos trabalhos mais modernos pode-se observar o uso de propostas híbridas que visam unir o melhor dos dois mundos fazendo-se valer das acuidades e velocidades das regras e utilizando as estatísticas para uma validação confirmadora. Os trabalhos

citados utilizam diversos idiomas, e algumas das soluções apresentadas podem ser aplicadas na língua portuguesa.

Verifica-se também que nenhuma das propostas encontradas na literatura faz uso de anotações advindas de estruturas *Multi-Terminal Multi-valued Decision Diagrams* (MTMDD) estruturas essas que permitem o uso de grande quantidade de dados de pesquisa, em formato de dicionários, por vezes multilíngue para a classificação de palavras de uma maneira extremamente rápida [6].

Com isto, este trabalho tem como motivação o fato de: (i) o problema de a ambiguidade ser um problema complexo e a proposta de novas técnicas para minimizar sua atuação é bem vinda; (ii) não existir propostas de desambiguação que utilizem estruturas MTMDD durante o *tagging* para a língua portuguesa; (iii) a possibilidade de se obter uma solução flexível, que por utilizar MTMDD, possa ser facilmente adaptada para novos idiomas, bastando que novos dicionários sejam criados, catalogados e adicionado à ferramenta.

2 CENÁRIO E CONTEXTUALIZAÇÃO

Neste capítulo serão apresentados conceitos fundamentais para a compressão deste trabalho. Na seção 2.1 será apresentada a definição para *Part-Of-Speech tagging*, sua importância e contribuição para a PLN. Na seção 2.2 serão descritos alguns tipos de *tagging*, que auxiliam na construção automática ou semiautomática de textos anotados. Na seção 2.3 será apresentado o conceito de desambiguação propriamente dito e as formas já apresentadas para lidar com esse problema. Na seção 2.4 serão apresentadas propostas de automatização de etiquetagem existentes, e suas diferenças. Por fim a seção 2.5 apresenta uma breve análise das propostas apresentadas.

2.1 Part-Of-Speech tagging

O grande objetivo de uma pesquisa em PLN é analisar e entender a linguagem. Como tarefas intermediárias a este objetivo final, temos objetivos menores que não requerem um completo entendimento de uma linguagem para ser realizado. Uma tarefa que pode se encaixar nessa descrição é a ação de etiquetar as palavras de um texto, catalogando-as ou simplesmente *Part-Of-Speech tagging* [8]. O termo *Part-Of-Speech* também pode ser entendido como a definição de classe

gramatical. Assim as palavras são anotadas de acordo com este conceito situado na morfologia que classifica cada palavra em um texto conforme sua distribuição sintática e morfológica. Essa classificação constitui na função executada por cada palavra em uma frase. Uma palavra anotada pode ser classificada como: substantivo, adjetivo, verbo, advérbio, preposição etc.

Na definição de Schmid [9], o *Part-Of-Speech tagging* é a tarefa que consiste em determinar corretamente as constituintes de uma sequencia de palavras. Neste prisma introduzem-se as dificuldades inerentes de palavras que podem agir de maneira distinta conforme o contexto empregado e também o complicado da ambiguidade típica que muitas palavras possuem. Um exemplo, em língua inglesa, que demostra as diversas facetas que podem ser observadas pela mesma palavra quando o contexto difere é apresentado a seguir. A palavra *back* será a palavra exemplo, que se apresenta como um substantivo na frase "*They stabbed him in the back*", um verbo no infinitivo em "*They will back the proposal*", um adjetivo na frase "*The charity owes $400,000 in back taxes*", e um advérbio na sentença "*It's too late to put the genie back in the bottle*."

Levando em consideração que a quantidade de *tags* assinaladas nos grupos de palavras embora possa flutuar bastante, de dúzias às centenas o mais comum é definir um grupo de etiquetas representativas que juntas somam algo entre 50 a 150 *tags*. Essa quantidade pode variar bastante de acordo com o idioma a ser utilizado bem como quão rico o idioma em questão é de regras morfossintáticas [9].

Schmid [9] comenta que diversos métodos já foram aplicados no processamento de POS *tagging* ao longo dos anos. Entre outras opções, foram utilizadas por Church [10] Cutting et. al. [11], e Brants [12] soluções com base em *Hidden Markov Models (HMM)*. Na década de 90, Brill [13] usou *transformation-based-learning*, Daelemans et al [14] utilizou *memory-based learning*, enquanto Ratnaparkhi [15] se valeu de *maximum-entropy modeling*. Outras opções utilizadas foram as redes neurais de Benello et al. [16] e Nakamura et al. [17], bem como as árvores de decisão empregadas por Black [18], Màrquez e Padró [19]. Podem ser encontradas ainda soluções que empregam *support vector machines [5] e* propostas de regras de desambiguação escritas de forma manual [20, 21, 22]. Ainda existe a possibilidade de se utilizar analisadores estatísticos para essa análise, no entanto os *taggers* mais comuns costumam ser muito mais rápidos e tendem a ser a opção mais acurada.

A construção de um *tagger* POS eficiente, como pode se imaginar, permite que sejam trilhados os mais diversos caminhos. Diversos trabalhos feitos na área apresentam uma releitura de trabalhos anteriores, utilizando uma técnica já empregada e reconhecidamente eficiente, no entanto adicionando alguma peculiaridade que agregue desempenho ou capacidade de maior acurácia. Dentro dessa gama de opções já citada anteriormente a seção a seguir tem como objetivo apresentar os tipos de *tagging* que constituem em opções interessantes para serem adaptadas como ferramentas do presente trabalho. Consequentemente, a construção da solução final, se dará apenas com os resultados dos testes realizados com os diversos tipos de *taggers* avaliando suas peculiaridades e analisando seus prós e contras, encontrando por fim uma solução que carregue o melhor custo-benefício.

2.2 Tipos de Tagging

Entre os diversos trabalhos e propostas já apresentados na área, serão brevemente comentados exemplos de processadores POST que se baseiam em ideias distintas na hora de resolver o problema de desambiguação das palavras. O primeiro utiliza-se de regras pré-definidas para fazer sua avaliação direta, o segundo utiliza estudos probabilísticos no momento de executar sua escolha pela melhor opção. Por último a forma que tem sido adotada por trabalhos recentes que se baseia em uma mescla dos dois tipos anteriores criando uma solução híbrida.

2.2.1 Part-Of-Speech tagging baseado em regras

Jurafsky e Martin [24] mostram que as soluções mais antigas (década de 60 e 70) de etiquetagem de Part-Of-Speech que se baseavam no uso de regras executavam duas etapas distintas. No primeiro momento é usado um dicionário que é capaz de listar todas as possibilidades válidas encontradas para cada palavra no texto. Em uma segunda etapa acontece a desambiguação que se vale de uma grande lusa de regras escritas manualmente. Com a ajuda dessa lista o sistema consegue filtrar as possibilidades encontradas, mantendo listas de possibilidades cada veze menores até a decisão por apenas uma etiqueta aconteça e a desambiguação termine. O comportamento encontrado nas soluções mais modernas não difere muito de seus predecessores. Como diferença evidente temos o tamanho dos dicionários atuais, bem como na quantidade de regras nas listas criadas para desambiguações disponíveis para uso hoje em dia. Baseado nessas fases de processamento, Karlsson [25] criou o *Constraint Grammar* (CG) uma metodologia dependente de contexto que sendo compilada em uma gramática é capaz de realizar a anotação de palavras de um texto. Esse trabalho se tornou uma base para diversas propostas que buscam adicionar funcionalidades ao CG.

Como exemplo de *tagger* baseado em CG e que segue esse paradigma temos o *EngCG*. Esta implementação específica trabalha como seus antepassados etiquetando cada palavra no texto com todas as possibilidades encontradas em seu dicionário em um primeiro momento. Após essa listagem e devida etiquetagem a segunda etapa sustentada por mais de 3700 regras de desambiguação começa a mineração das opções existentes, eliminando as indicações inválidas culminando em um texto anotado adequadamente com apenas uma *tag* para cada palavra, tendo o objetivo de sustentar a melhor opção em cada desambiguação. As regras podem ser realmente complexas, levando em consideração diversas situações que encadeadas culminam em adições ou remoções de etiquetas durante o processamento. Na Figura 1 pode-se observar uma regra simplificada com alguns argumentos, utilizada quando no texto se encontra a palavra "*that*".

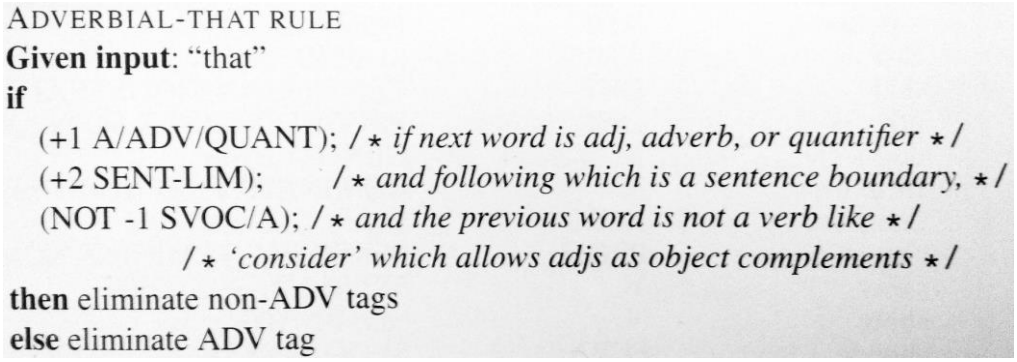
ADVERBIAL-THAT RULE
Given input: "that"
if
(+1 A/ADV/QUANT); /* *if next word is adj, adverb, or quantifier* */
(+2 SENT-LIM); /* *and following which is a sentence boundary,* */
(NOT -1 SVOC/A); /* *and the previous word is not a verb like* */
/* *'consider' which allows adjs as object complements* */
then eliminate non-ADV tags
else eliminate ADV tag

Figura 1- Regra relativa à palavra "*that*" no idioma inglês.
Fonte: Jurafsky e Martin [24]

Nota-se que uma regra pode ser formada com diversas premissas necessárias e com variações de comportamento, para uma situação de casamento das premissas ou uma situação adversa. Os trabalhos de Aduriz e Illarraza [3] e Brill [4] se encaixam nessa área de estudo.

2.2.2 Part-Of-Speech tagging usando HMM

A intenção de se utilizar estatística como parte da solução de POST, não é uma ideia nova. O uso de probabilidades foi visto na década de 60 em um trabalho de Stolz [26]. Na década de 70, Bahl e Mercer [27] apresentaram um protótipo de *tagger* probabilístico que utilizada decodificação através

do algoritmo de Viterbi[1]. Durante os anos 80 diversos *taggers* estocásticos foram criados, Church [10] e DeRose [29] foram alguns dos autores.

Uma das soluções mais vistas em trabalhos de POST baseado em probabilidade é o uso de *Hidden Markov Models* como sendo o algoritmo estocástico de etiquetagem. Jurafsky e Martin definem a utilização de HMM como sendo um caso especial de inferência Bayseana também conhecido por classificação Bayseana, paradigma que foi originalmente conhecido através do trabalho de Bayes [30]. Utilizando uma explicação simplificada, essa inferência busca encontrar em uma sequência de palavras, como em uma frase, quais são as *tags* que são as mais corretas, partindo de uma etapa inicial onde todas as *tags* são válidas. Diversas fórmulas são utilizadas para encontrar o objetivo que através desse paradigma, no entanto a utilização HMM traz consigo duas premissas facilitadoras.

- A probabilidade de uma palavra aparecer é determinada somente por sua própria POS *tag*, isto é, essa possibilidade é independente das palavras e *tags* ao seu redor.

- A probabilidade de uma *tag* aparecer no texto é dependente apenas da *tag* imediatamente anterior, e não leva em consideração a sequência completa das *tags.*

Com as premissas apresentadas, uma nova fórmula baseada nas teorias de Bayes pode ser obtida. A fórmula em questão, que pode ser observada na figura 2, define uma equação onde um etiquetador de bigramas, é capaz de estimar a mais provável sequência de *tags.*

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n|w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

Figura 2 - Equação de HMM com dois tipos de probabilidades, transição de *tags* e possibilidade de palavras.
Fonte: Jurafsky e Martin [24].

Como exemplos desse método podem ser citadas as propostas de Giménez e Màrquez [5] e Segond et.al. [7].

[1] Viterbi é um algoritmo de programação dinâmica que tem como objetivo encontrar o "Viterbi Path" que seria uma determinada sequencia de estados ocultos. Possui esse nome devido ao seu criador Andrew Viterbi [28].

Explicando um pouco melhor a solução apresentada em [5] temos um pacote de ferramentas que se baseiam no uso de *Support Vector Machines* (SVM). Esse pacote consiste em três componentes principais divididos a partir das funções por eles executadas: SVMlearn (componente de aprendizado), SVMTagger (o etiquetador) e por fim o SVMeval, componente este responsável pela avaliação dos resultados obtidos pelos outro dois. Focando exclusivamente na função *tagger* temos uma solução robusta que percorre linha a linha do corpus anotando as palavras, levando em consideração para a palavra seguinte, a informação obtida sobre a palavra atual. Demonstrando flexibilidade, o SVMtagger pode ser configurado para avaliar as palavras em contexto reduzido, ou a nível de sentença. As direções que o *tagger* percorre também podem ser ajustadas, resolvendo da "direita para a esquerda" e vice-versa. Entre outras configurações uma opção importante é a utilização de múltiplos passos de etiquetagem o que embora envolva mais tempo, normalmente realiza uma desambiguação de melhor qualidade final.

O trabalho apresentado em [6], no entanto, buscou utilizar o HMM em seu formato clássico formulando uma desambiguação de palavras totalmente baseada em probabilidade. Este trabalho usou como contexto o formato apresentado anteriormente baseado em bigramas. Desta forma as cadeias são montadas de maneira onde apenas o estado imediatamente anterior possui relevância. Para ajudar a compreender e visualizar como isso funciona, é interessante comentar como a criação das Cadeias de Markov (*Markov Chains*) ocorre. Como explicação trivial, pode-se dizer que a cadeia de Markov pode ser vista como um autômato finito, que possui transições acionadas pela ocorrência de processos estocásticos [32]. Embora os modelos markovianos possam ser encontrados nas mais diversas combinações (Stochastic Automata Networks, HMM , etc.) existem apenas dois tipos de cadeias de Markov:

- CTMC - *Continuous Time Markov Chains*
- DTMC - *Discrete Time Markov Chains*

Sua principal diferença consiste na forma em que cada tipo descreve uma transição entre os estados possíveis da cadeia. Enquanto o CTMC descreve cada transição entre estados como taxas de ocorrência, o DTMC apresenta a transição como probabilidade. Partindo do pressuposto que as cadeias de Markov tiveram como base as ideias de Bayes, sabemos que a soma das N possibilidades existentes de transições em um DTMC deve ter seu valor somado igual ou inferior a 1.

Exemplificando graficamente uma simples cadeia de Markov do tipo DTMC, temos a figura 3.

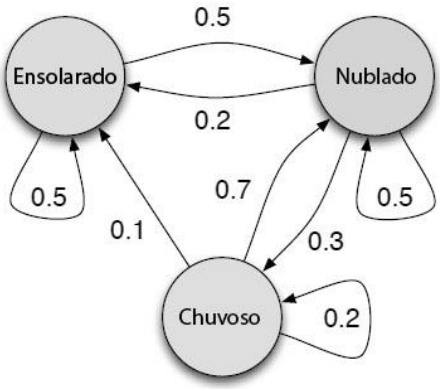

Figura 3 - Cadeia de Markov simples representando possibilidades climáticas.

Condicionado à ideia de que cada aresta representa em valor probabilístico de cada estado transacionando para outro temos as seguintes inferências. Um clima chuvoso possui uma probabilidade de 0.2 de continuar assim, enquanto tem 0.7 de chance de tornar-se nublado. Desta forma resta a probabilidade 0.1 de que o clima atualmente chuvoso torne-se ensolarado. Seguindo essa ideia podem-se averiguar as possibilidades validas de cada estado e a chance de cada transição acontecer. Uma grande característica das cadeias markovianas representada no modelo é a ausência de memória. Também conhecida como propriedade markoviana essa condição faz com que apenas o estado atual seja relevante no momento do cálculo da possibilidade futura, não interessando também a quantidade de tempo atualmente utilizado no estado corrente.

No caso específico do trabalho [6] foi utilizado um tagger mais antigo, de 1992r baseado em HMM e descrito formalmente no trabalho de Cutting et. al. [32]. Este por sua vez apresenta o formalismo do modelo e quais algoritmos são geralmente utilizados para a obtenção de estimativas, sendo comumente usados ou a proposta de Viterbi ou o trabalho de Baum Welch, solução apresentada em [32].

Entrando no detalhe de ambas as soluções percebemos que esses aplicativos visam resolver o problema da desambiguação através de um grupo de HMMs que são geradas automaticamente na fase de aprendizagem do *tagger*, fase essa que ocorre normalmente como uma prévia dos testes reais. Neste momento é que os ajustes finos da aplicação podem ser efetuados resultando assim em uma solução mais acurada dentro de um determinado tema textual. Para manter um bom nível de

eficiência os *taggers* costumam ser treinados com textos de mesma área de conhecimento daqueles que serão utilizados nos futuros testes. Isso ajuda a catalogar as virtudes e deficiências de uma determinada solução bem como facilita a comparação desta com as demais que normalmente também se focam em um determinado nicho textual.

2.2.3 Part-Of-Speech tagging híbrido

Para melhorar o desempenho de acerto (palavras corretamente etiquetadas), normalmente menor dos métodos estatísticos e também conseguir ser mais específico quando trabalhando com um idioma mais complexo, diversos autores partiram para a utilização de um método híbrido. Este método que visa pegar o melhor dos dois mundos, o da estatística e o das regras, concentra as melhores facetas de cada método tentando assim elaborar uma solução mais eficiente daquela alcança com apenas um paradigma [31].

2.3 Desambiguação

No momento que lemos um texto, podemos nos deparar com diversas situações onde uma determinada palavra ou conjunto de palavras podem possuir significados distintos. Essa desambiguação acontece normalmente durante uma leitura já que estamos com o contexto das palavras em nossa memória. No entanto se apenas um conjunto de palavras com ambiguidade forem apresentadas sem um contexto adequado, escolher a melhor opção de significado se torna uma tarefa árdua [23].

Dentro de cada formato de POS *tagger* utilizado, uma maneira diferente é encontrada para lidar com esse problema. No caso dos *taggers* probabilísticos, estes fazem uso basicamente do algoritmo de Viterbi para desambiguar as palavras que acabam possuindo mais de uma *tag* após a análise superficial do texto (*shallow parsing*). Enquanto os *taggers* que possuem um processamento baseado no uso de regras pré-determinadas, aprendidas ou inseridas manualmente, utilizam-se dessa mineração de possibilidades relevantes que são filtradas em cada iteração. Essa etapa de mineração dos dados, através das regras termina quando apenas uma *tag* é escolhida para ser anotada em cada palavra do texto. Existem ainda os formatos utilizados pelas soluções híbridas que fundem algumas etapas para tentar obter uma taxa de acerto mais elevada do que as soluções originais. Estudos como o de Tapanainen e Voutilainen [21] já alcançaram até 98% de taxa de acerto utilizando métodos híbridos.

2.4 Trabalhos Similares

Os trabalhos similares buscados foram aqueles que conseguiram em seus diversos métodos bons resultados em relação às técnicas mais conhecidas. Ainda, para ter uma boa visão das possibilidades que essa área de estudo apresenta, foram escolhidos trabalhos que usem maneiras de POS *tagging* baseadas em probabilidades ou que utilizem regras de desambiguação.

Como um dos artigos base da área que utiliza desambiguação através de regras foi escolhido o trabalho de Brill [4]. Essa proposta descreve um algoritmo que é capaz de realizar aprendizagem sem supervisão, aprendendo assim a partir de corpus não anotados manualmente. Neste trabalho ele as também apresenta uma integração das possibilidades de treinamento para a obtenção de regras, combinando algoritmos de treinamento supervisionado e não supervisionado criando assim um *tagger* de alto desempenho com uma necessidade pequena de textos pré-anotados.

Seguindo na linha de desambiguação baseada em regras, o trabalho de Aduriz e Illarraza [3], aparece como um trabalho onde uma análise de ambiguidades morfossintáticas é realizada, sendo focado especialmente no idioma basco. Essa proposta utiliza como *parser* o já comentado *Constraint Grammar* (CG) [25] que já foi diversas vezes utilizado na criação de gramáticas para vários idiomas diferentes. Sua colaboração foi apresentar um resultado de melhoria nas tarefas de identificação e ambiguidades em funções sintáticas e morfossintáticas utilizando uma análise superficial. Para em seguida realizar as desambiguações através das mil regras geradas para este fim.

Exemplo de um representante da linha de pesquisa de *taggers* estatísticos, o trabalho de Giménez e Màrquez [5] apresenta uma proposta de POS tagger baseado em *Support Vector Machines* chamado de *SVMTool*. Essa ferramenta se apresenta como uma opção simples, flexível, e eficiente para as necessidades atuais do processamento de linguagem natural. De acordo com os autores a *SVMTool* simples de utilizar necessitando de poucos parâmetros para funcionar em seu formato na linguagem *Pearl*. O contexto a ser adotado na ferramenta também é passível de ajuste, definindo tamanhos dos N-gramas (bigramas, trigramas, etc.), podendo ajustar também o tempo de *tagging* a ser executado. Outra vantagem apresentada no trabalho é que a ferramenta tende a se comportar bem não importando o idioma utilizado. Nos testes, utilizando tanto o inglês quanto o espanhol foi possível observar marcas consideráveis nas taxas de globais, (96,16% e 96,89% respectivamente). Para tanto além de uma etapa de aprendizagem não supervisionada de textos no

idioma pretendido, é necessário adicionar às suas configurações dicionários morfossintáticos compatíveis com estas linguagens.

Mantendo as propostas baseadas em probabilidades, Segond et.al.[7] apresenta um experimento que através de HMM consiste em uma formatação clássica de um *tagger* deste estilo. A preparação de dados se dá com a obtenção de todas as *tags* semânticas possíveis obtidas na *WordNet*[2]. Em seguida, tendo como origem o *Brown Corpus,* foi gerados um *corpus* de treinamento e um corpus para os testes propriamente. Nos testes o HMM utilizado apenas era capaz de interpretar bigramas, isto é cada palavra apenas leva em consideração a palavra imediatamente anterior. Por fim realizado o processamento do *corpus* de treino a fim de realizar os ajustes necessários no algoritmo de *tagging,* foi processado o *corpus* de teste. Na sequência foi realizada uma comparação das *tags* encontradas nos no processamento, com as já existentes no conjunto de palavras do treinamento inicial que já haviam sido etiquetadas manualmente. Três testes distintos foram realizados, conseguindo uma taxa máxima de acerto global de 89%. Como possibilidade de melhoria o trabalho conclui que adicionar a capacidade de consulta a dicionários sintáticos poderia incrementar o desempenho obtido através da formatação clássica executada.

Como exemplo a ser destacado, com um formato já ajustado e adequado à utilização do idioma português tem a tese de Domingues [34] que discursa sobre uma abordagem completa para o desenvolvimento de um etiquetador de alta acurácia para o português do Brasil. Esse estudo exploratório apresenta solução que foi idealizada como uma visão híbrida combinando etiquetagem probabilística e etiquetagem baseada em regras. Foram utilizados quatro versões dos seguintes corpora CETENFolha, Bosque CF 7.4, Mac-Morpho e Selva Científica. A solução de tagging foi utilizar ferramentas open source já existentes, entre elas o gerador de regras automatizado μ-TBL (Micro transformation-based learning) e o TreeTagger como solução de tagging. Sua solução final apresenta diversas fases de processamentos, que começa com a tokenização do texto, passando pelo parser estatístico. Após essa etapa ocorre a consulta por nomes próprios no texto e por fim a etiquetagem baseada em regras. Sendo esta última etapa diferenciada, pois se utiliza de três grupos de regras. Todo esse processo permitiu que os experimentos com melhor desempenho atingissem uma acurácia global superior a 98%.

[2] WordNet é um grande banco de dados com dados léxicos no idioma inglês. Cada um expressando um conceito, substantivos, verbos, adjetivos e advérbios são agrupados em conjuntos de sinônimos cognitivos, os *synsets.*

2.5 Análise dos Trabalhos Similares

Para sintetizar as informações coletadas nos trabalhos similares apresentados na seção anterior, as principais características relevantes para este trabalho foram organizadas em forma de tabela e são apresentadas na Tabela 1. Esta análise, todavia não possui um caráter de escolha entre as propostas a seguir apresentadas. Servem, no entanto para obter um entendimento maior sobre o problema. Desta forma o a continua observação dos trabalhos já utilizando as ferramentas adequadas para este problema, colabora na compreensão ao produzir subsídios necessários para a escolha de um próprio método de desambiguação. Esse método próprio pode vislumbrar algo novo, aplicar soluções realizadas em outros idiomas tornando-o útil para textos em português, ou ainda conseguir uma mescla de ambas as possibilidades.

Tabela 1. Comparação entre trabalhos similares

Identificação da Proposta	**Tipo de *tagging***	**Máxima acurácia obtida em testes?**	**Em qual língua é aplicada?**	**Tipo de desambiguação utilizado**
Aduriz e Illarraza [3]	Baseado em regras, usando uma gramática própria.	97.51%	Basco	Regras de desambiguação
Brill [4]	Baseado em regras, com aprendizagem sem supervisão. (Transformation-Based Learning)	96%	Inglês	Regras de desambiguação
Giménez e Màrquez [5]	Utilizando *Support Vector Machines* (SVM)	97.16%	Inglês, Espanhol	Estatístico
Segond et.al.[7]	Baseado em HMM	89%	Inglês	Estatístico
Domingues [34]	Processo híbrido baseado em probabilidade e regras	98,30%	Português do Brasil	Estatístico/Regras

3 PROPOSTA PARA DISSERTAÇÃO DE MESTRADO

Neste capítulo será apresentada a proposta para Dissertação de Mestrado. Como partes da proposta serão apresentadas os objetivos deste trabalho e a metodologia, que descreverá de forma breve os passos que serão dados para alcançar os objetivos propostos. Ainda como parte da proposta será descrito o cronograma de atividades, dando uma breve visão de como o trabalho será conduzido com relação ao tempo.

3.1 Objetivos

O objetivo geral do trabalho é propor um desambiguador estocasticamente eficiente de anotações morfossintáticas obtidas via MTMDD. Especificamente a solução tem foco na desambiguização de classes gramaticais em textos da língua portuguesa. A fim de alcançar um nível de acuidade semelhante aos trabalhos similares apresentados, este projeto realizará uma trajetória de melhoria contínua. Desta maneira busca-se utilizar o grande desempenho encontrado nos programas que se valem de estruturas MTMDD a fim de criar um desambiguador ao mesmo tempo altamente confiável e preciso quanto veloz. Começando de uma maneira simples, desambiguando trivialmente, para em seguida melhorar este processo já visando uma nova melhora no futuro. Agregando essa ideia de desenvolvimento ágil, tem-se o intuito de recriar as etapas base que as soluções agregando novas capacidades na expectativa da melhoria contínua dos resultados obtidos. Conforme já comentado anteriormente de acordo com a capacidade das ferramentas apresentadas como base do trabalho, a possibilidade de adaptação para outros idiomas no futuro é factível. Para que este objetivo seja alcançado, foram definidos os seguintes objetivos específicos:

I. Criar um *tagger* probabilístico que seja capaz de desambiguar de maneira aleatória a classe gramatical entre as múltiplas possibilidades encontradas no MTMDD.

II. Implementar uma melhoria desse *tagger* partindo de uma lista de utilização das classes gramaticais mais comuns para cada palavra. Um pré-processamento do corpus listará para cada palavra, quais as classes gramaticais encontradas obtendo assim um índice percentual que será usado na desambiguação. A escolha neste caso se dará utilizando esses valores como base em cada avaliação de palavra anotada múltiplas vezes.

III. Implementar uma nova melhoria desse *tagger* utilizando no cálculo os valores das classes gramaticais das palavras vizinhas à que está sendo avaliada. Essa proposta agrega mais informação e consequentemente tem como objetivo uma acuidade maior;

IV. Durante esse processo, colaborar com professores e colegas a fim de encontrar oportunidades de melhorias tanto dessa nova ferramenta proposta, quanto soluções já existentes como o etiquetador para MTMDD.

V. Apresentar possibilidade de melhoria e eventualmente implementar uma maneira de adicionar a capacidade de resolução de regras ao *tagger* a fim de fazer uma revalidação após a etiquetagem probabilística.

VI. Gerar resultados das análises da solução proposta perante soluções já conhecidas.

Para que os objetivos traçados nesta proposta sejam concluídos, na seção seguinte será descrita a metodologia que será utilizada.

3.2 Metodologia

Observou-se nos trabalhos similares que já havia uma escolha realizada a respeito de qual formato de *tagger* que seria o escolhido na proposta. Neste trabalho, no entanto, os resultados obtidos nos protótipos criados em cada fase farão com que se busque continuamente a escolha mais adequada. Partindo de uma solução simples, aumentando sua complexidade ao longo do tempo com a expectativa de melhoria no desempenho, podemos ter o embasamento necessário de como construir um desambiguador desde o começo. Esse tipo de abordagem nos permite uma ter uma visão clara e concreta, baseada em fatos, de quais serão as melhores escolhas para adicionar a este trabalho.

A primeira etapa do trabalho consistirá em criar ferramentas próprias para ajustar um conjunto de *corpus* que será utilizado para a avaliação. O conjunto de corpus deve se ater aos mais conhecidos como CETENFolha e Mac-Morpho. Podendo se valer da característica do Mac-Morpho de ser um corpus com etiquetagem revisada a avaliação da qualidade do parser será medida com o seu comparativo com o texto previamente anotado. O intuito do processo de melhoria contínua é adicionar novas funcionalidades e escopo aos poucos, assim para cada novo processo criado um

melhor desempenho do *tagger* é esperado. Idealmente a ferramenta deve chegar ao patamar de agregar funcionalidades híbridas sendo estas estatísticas e baseadas em regras. Deverá ser criada uma lista de necessidades relevantes, para que os *taggers* criados possam ser melhorados ao decorrer do processo de criação.

Como soluções funcionais, os *taggers* serão testados com os *corpora* previamente obtidos a fim de que se possa criar uma classificação destes a partir do desempenho obtido através dos testes. Essa análise será crucial para rever configurações, realizar ajustes e adequações pontuais. Durante esse período deverá ser apresentada em forma de seminário, a atual situação do andamento do trabalho.

Após os diversos testes e obtendo valores adequados para cada tipo de abordagem, seguir até onde o cronograma permitir com novos ajustes e melhorias pontuais a fim de gerar uma proposta que consiga trazer de a desambiguação adequada para textos que foram previamente anotados através de dicionários MTMDD existentes. Durante esse longo processo o texto final da dissertação também estará sendo redigido. Obtendo resultados relevantes em relação às opções disponíveis será interessante redigir artigos científicos para submissão em periódicos desta área de pesquisa. No final do ano de 2014 com o trabalho completo, será o momento de defender a dissertação perante a banca julgadora designada.

3.3 Cronograma de Atividades

A Tabela 4 apresenta o cronograma que organiza temporalmente as atividades que serão desenvolvidas para a realização do trabalho apresentado nesta proposta durante o ano de 2014. As atividades estão identificadas na Tabela 4 por números e fazem referência aos seguintes passos:

1. Criar uma solução de *tagger* que seja capaz de desambiguar classes gramaticais computadas via MTMDD utilizando aleatoriedade;

2. Melhorar a versão anterior adicionando ao cálculo de desambiguação o percentual de cada classe obtido em uma fase de pré-avaliação do corpus.

3. Implementar um algoritmo que leve em consideração a possibilidade das palavras vizinhas no cálculo de desambiguação de uma determinada palavra.

4. Apresentar melhorias passíveis de implementação agregando funcionalidades à solução obtida até o momento.

5. Realizar avaliação da solução construída, de modo a comparar com trabalhos semelhantes e buscar entender como melhorar seu desempenho;

6. Propor e uma solução final do desambiguador, agregando as boas práticas encontradas nos trabalhos correlatos bem como melhorias que surgirem após a etapa de testes;

7. Redigir o texto da Dissertação de Mestrado;

8. Redigir artigos científicos para submissão em congressos ou periódicos da área;

9. Defender a Dissertação de Mestrado.

Tabela 2. Cronograma de atividades

Atividade	jul/14	ago/14	set/14	out/14	nov/14	dez/14
1	X					
2		X				
3			X			
4				X		
5				X		
6					X	
7			X	X	X	
8						X
9						X

REFERÊNCIAS BIBLIOGRÁFICAS

[1] SINTEF, *Big Data, for better or worse: 90% of world's data generated over last two years*, In ScienceDaily. Disponível em: <http://www.sciencedaily.com/releases/2013/05/130522085217.htm>. Acesso em: 12 de dezembro de 2013.

[2] SILVA, JOÃO RICARDO, *Shallow Processing of Portuguese: From Sentence Chunking to Nominal Lemmatization*, In Master's thesis, Lisboa University, Portugal. 2007

[3] ADURIZ ITZIAR, ILLARRAZA ARANTA DÍAS DE, *Morphosyntactic disambiguation and shallow parsing in computational processing of Basque*. In Inquiries into the lexicon-syntax relations in Basque / Bernard Oyharçabal (aut.), 2004, ISBN 84-8373-580-6, pp. 1-23.

[4] BRILL ERIC, *Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging*. In Natural Language Processing Using Very Large Corpora. Kluwer Academic Press, 1995, pp. 1-13.

[5] GIMÉNEZ JESÚS, MÀRQUEZ LLUÍS, *Svmtool: A general pos tagger generator based on support vector machines*. In Proceedings of the 4th International Conference on Language Resources and Evaluation, 2004, pp. 43-46.

[6] FERNANDES PAULO, LOPES LUCELENE, PROLO CARLOS AUGUSTO, SALES AFONSO, VIEIRA RENATA, *A Fast, Memory Efficient, Scalable and Multilingual Dictionary Retriever*. In LREC, 2012, pp. 2520-2524.

[7] SEGOND FREDERIQUE, SCHILLER ANNE, GREFENSTETTE GREGORY, CHANOD JEAN-PIERRE, *An Experiment In Semantic Tagging Using Hidden Markov Model Tagging*. In ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, 1997, pp. 78-81.

[8] MANNING CHRIS, SCHÜTZE HINRICH, *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge. 1999, pp. 341-380

[9] SCHMID, HELMUT, *Part-Of-Speech Tagging Reference.* In European Summer School in Logic, Language and Information. Disponível em: <http://www.coli.uni-sa ar lan d.de/~schulte /Tea ching/ESSLLI-06/Referenzen/Tagging/schmid-hsk-tag.pdf>. Acessado em: 11 de dezembro de 2013 .

[10] CHURCH, K. W. *A stochastic parts program and noun phrase parser for unrestricted text*. In Proceedings of the Second Conference on Applied Natural Language Processing, 1988, pp.136–143.

[11] CUTTING D., KUPIEC J., PEDERSEN J., SIBUN P. *A practical part-of-speech tagger*. In Proceedings of the Third Conference on Applied Natural Language Processing, 1994, pp. 133–140

[12] BRANTS, T., *TnT - a statistical part-of-speech tagger*. In Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000, 2000.

[13] BRILL, E., *A simple rule-based part of speech tagger*. In Proceedings of the Third Conference on Applied Natural Language Processing, 1992.

[14] DAELEMANS W., ZAVREL J., BERCK P., GILLIS S., *Mbt: A memory-based part of speech tagger-generator*. In Ejerhed, E. and Dagan, I., editors, Proceedings of the Fourth Workshop on Very Large Corpora, 1996, pp. 14–27.

[15] RATNAPARKHI A., *A maximum entropy model for part-of-speech tagging*. In Proceedings of Conference on Empirical Methods in Natural Language Processing. University of Pennsylvania, 1996.

[16] BENELLO J., MACKIE A. W., ERSON J. A., *Syntactic category disambiguation with neural networks*. Computer Speech and Language, 1989, Vol. 3, pp. 203–217.

[17] NAKAMURA M., MARUYAMA K., KAWABATA T., SHIKANO K., *Neural network approach to word category prediction for English texts*. In Karlgren, H., editor, Proceedings of the 13th International Conference on Computational Linguistics COLING 1990, 1990, pp. 213–218.

[18] BLACK E., JELINEK F., LAFFERTY J., MERCER R., ROUKOS S., *Decision tree models applied to the labeling of text with parts of speech*. In Proceedings of the DARPA Speech and Natural Language Workshop, 1992.

[19] MÀRQUEZ L., PADRÓ L., *A flexible POS tagger using an automatically acquired language model*. In Proceedings of the 35th Annual Meeting of the ACL, 1997, pp. 238– 245.

[20] GREENE B., RUBIN G., *Automatic grammatical tagging of English*. Technical report, Department of Linguistics, Brown University, 1971.

[21] Tapanainen P., Voutilainen A., *Tagging accurately - don't guess if you know*. In Proceedings of the 4th Conference on Applied Natural Language Processing, 1994, pp. 47–52.

[22] KLATT, S., *Combining a rule-based tagger with a statistical tagger for annotating German texts*. In Buseman, S., editor, KONVENS 2002, 2002.

[23] MARTINS, PEDRO LOPES MENDES, *Desambiguação Automática da Flexão Verbal em Contexto*. In Master's thesis, Lisboa University, Portugal. 2008.

[24] JURAFSKY DANIEL, MARTIN JAMES, *Speech and Language Processing: An Introduction to Natural Language Processing*, Computational Linguistics, and Speech Recognition, Prentice Hall, 2008.

[25] KARLSSON FRED, *Constraint Grammar as a Framework for Parsing Unrestricted Text*. H. Karlgren, ed., In Proceedings of the 13th International Conference of Computational Linguistics, Vol. 3. Helsinki 1990, pp. 168-173.

[26] STOLZ WALTER, TANNENBAUM PERCY, CARSTENSEN FRÉDERICK, *A Stochastic approach to the grammatical coding of English*. Communications of the ACM, 1965, pp. 399-405

[27] BAHL L., MERCER R., *Part-Of-Speech assignment by a statistical decision algorithm*. In symposium on Information Theory, 1976, pp. 88-89.

[28] VITERBI ANDREW, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*. IEEE Transactions on Information Theory 13, 1967, pp. 260–269.

[29] DEROSE STEVEN, *Grammatical Category Disambiguation by Statistical Optimization*. Computational Linguistics Vol. 14, 1988, pp. 31-39

[30] BAYES T., *An essay towards solving a problem in the doctrine of chances*. Phil. Trans. of the Royal Soc. of London, 1763, pp. 370-418

[31] NAKAGAWA T., UCHIMOTO K., *A hybrid approach to word segmentation and pos tagging*. Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ser. ACL '07, 2007, pp. 217–220.

[32] W. J. STEWART., *Probability, Markov Chains, Queues, and Simulation.* Princeton University Press, USA, 2009.

[32] DOUG CUTTING, JULIAN KUPIEC, JAN PEDERSEN, PENELOPE SIBUN. *A Practical Part-of- speech Tagger*. Proceedmgs of ANLP-92. Trento, Italy, 1992.

[33] L. E. BAUM. *An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process.* Inequalities, 3:1-8, 1972.

[34] DOMINGUES, MIRIAM LÚCIA CAMPOS SERRA. *Abordagem para o desenvolvimento de um etiquetador de alta acurácia para o Português do Brasil*. 2011. 140 f. Tese (Doutorado) – Universidade Federal do Pará, Instituto de Tecnologia, Belém, 2011. Programa de Pós-Graduação em Engenharia Elétrica.