FACULDADE DE INFORMÁTICA
PUCRS – Brazil
http://www.inf.pucrs.br

Fundamentos do Processamento Estatístico da Linguagem Natural

Caroline Varaschin Gasperin e Vera Lúcia Strube de Lima

TECHNICAL REPORT SERIES

Number 021
November, 2001

Contact:

caroline@inf.pucrs.br

http://www.inf.pucrs.br/~caroline

vera@inf.pucrs.br

http://www.inf.pucrs.br/~vera

Caroline Varaschin Gasperin is a graduate student of PPGCC at PUCRS/Brazil. She is a member of the FUNDAÇÕES research project since 2000. She develops research in natural language processing, applied to information retrieval. She receives a federal graduate research grant from CAPES (Brazil) to support her research.

Vera Lúcia Strube de Lima works at PUCRS/Brazil since 1979. She is a titular professor and coordinator of the FUNDAÇÕES and CONTEXTO research projects (grants from CAPES and CNPq – Brazil). She develops research in natural language processing. She got her Ph.D. in 1990 at Université Joseph Fourier (Grenoble, France).

Published by PPGCC – FACIN – PUCRS
Av. Ipiranga, 6681
90619-900 Porto Alegre – RS – Brazil

Fundamentos do Processamento Estatístico da Linguagem Natural

Relatório Técnico N° 021/2001

Caroline Varaschin Gasperin Vera Lúcia Strube de Lima

1 Introdução

A linguagem é um dos aspectos fundamentais do comportamento humano, pois permite a interação entre os indivíduos e a perpetuação dos conhecimentos. O ser humano compreende e desenvolve de forma natural sentenças na linguagem que aprendeu desde criança.

O processamento automático da linguagem natural (PLN) visa aproximar o computador da realidade do homem, através do desenvolvimento de ferramentas que possibilitem uma comunicação mais natural entre homem e máquina, além de ferramentas para a extração de informações de grandes bases textuais, tradução automática de textos, etc.

Nos últimos anos, com a disponibilização de grandes bases de textos em formato digital, foram sendo difundidas técnicas baseadas em conceitos estatísticos para fazer a análise destes textos.

O processamento estatístico da linguagem natural consiste no uso de abordagens quantitativas para o processamento automático de textos. Uma coleta de informações é realizada sobre uma grande base de textos, denominada corpus, para o levantamento das probabilidades de ocorrência de palavras, seqüências de palavras, categorias de palavras, etc.

Existem diferentes tipos de corpora, que contêm diferentes tipos e quantidades de textos. O corpus utilizado deve ser representativo, contendo o máximo possível das palavras e estruturas existentes na linguagem. No entanto, de acordo com a mais conhecida lei de Zipf [MAN99], um problema da linguagem natural é a esparcidade dos dados: há poucas palavras muito comuns, há um número médio de palavras com freqüência média, e há muitas palavras com baixa freqüência.

Com o uso de métodos estatísticos é possível, conforme [VIL95], fazer a análise de textos irrestritos, ou seja, textos sem restrições em relação ao formato, tamanho, ou estruturas lingüísticas presentes.

A partir de um corpus, pode-se obter informações sobre as palavras e sobre a estrutura da linguagem. Observando-se a freqüência de co-ocorrência das palavras, podem ser descobertas as "colocações" existentes em um corpus, as quais são expressões formadas por palavras que, quando aparecem juntas, têm um significado especial. Observando-se o comportamento dos lexemas[1] no corpus, pode-se estimar o comportamento dos lexemas para a toda a linguagem. Propriedades das palavras possibilitam a aquisição de conhecimento lexical, como restrições de complementos e classes dos lexemas. Ainda, pode-se estimar o significado de palavras desconhecidas com base em propriedades de palavras conhecidas similares (estabelece-se uma medida de similaridade).

Em relação à estrutura da linguagem, o processamento estatístico também possibilita o tratamento da sintaxe das sentenças do corpus. Dentre os principais problemas a serem tratados no PLN está a questão da ambigüidade, que pode ser sintática ou semântica. A ambigüidade sintática consiste na existência de mais de uma estrutura sintática para uma mesma sentença. Através de métodos estatísticos, o problema da ambigüidade pode ser contornado. A ambigüidade de ligação, que é um tipo de ambigüidade sintática e consiste em determinar a que elemento da sentença está ligada uma expressão ambígüa, pode ser resolvida através da aquisição de propriedades léxicas das palavras a partir do corpus utilizado. Outros tipos de ambigüidade sintática podem ser resolvidos através da marcação de textos utilizando-se modelos estatísticos, como os baseados no modelo

[1] Lexema consiste em uma única entrada no dicionário, com um único significado.

de Markov. Estes modelos selecionam a melhor seqüência de rótulos para as palavras de uma determinada sentença, assim decidindo a estrutura sintática mais adequada. A ambigüidade sintática também pode ser resolvida através da análise sintática da sentença, de acordo com uma gramática livre de contexto probabilística - PCFG. As PCFGs possuem probabilidades associadas às suas regras, que servem para fazer um *ranking* entre as diferentes estruturas sintáticas possíveis para uma sentença.

A ambigüidade semântica consiste na existência de mais de um significado para uma mesma palavra. A resolução da ambigüidade semântica é útil em sistemas de recuperação de informações. Esta questão pode ser tratada por diferentes métodos estatísticos: uns supervisonados, que utilizam corpora marcados, outros baseados em dicionários ou thesauri, e ainda métodos não supervisionados, que utilizam corpora não marcados.

O objetivo do estudo relatado neste trabalho é fazer um levantamento dos modelos estatísticos existentes para o tratamento dos diferentes aspectos do PLN, nos diversos níveis do processamento. Com isso, pretende-se compor um documento que sirva como fonte de pesquisa a pessoas interessadas em conhecer a abordagem estatística do PLN, principalmente porque não foi encontrado material sobre o assunto em português.

A estrutura deste trabalho é semelhante à estrutura de [MAN99], pois a ordem com que os tópicos foram abordados neste livro foi considerada bastante intuitiva, favorecendo o entendimento e relacionamento das técnicas apresentadas. No entanto, em cada tópico, são mesclados os dados obtidos nas demais fontes de pesquisa para este levantamento.

O presente trabalho está organizado em 7 seções, sendo a primeira esta introdução.

Na seção 2 deste trabalho, são introduzidos os principais conceitos sobre a teoria da probabilidade e a teoria da informação, que serão necessários para o entendimento das seções seguintes.

A seção 3 trata das características de um corpus, das condições necessárias a seu processamento, e das questões relacionadas a sua forma de utilização.

A seção 4 explora os modelos estatísticos para obtenção de conhecimento a partir das palavras de um texto. Nesta seção, são apresentados métodos para detecção de colocações, para inferência estatística de propriedades da linguagem, para redução da ambigüidade semântica, e para aquisição de dados léxicos.

Na seção 5, são apresentados os modelos estatísticos para obtenção de conhecimentos sintáticos sobre a linguagem. Aqui, são apresentados a teoria dos modelos de Markov, modelos para marcação de categorias das palavras, gramáticas probabilísticas e questões sobre a análise sintática probabilística.

Na seção 6, são apresentadas aplicações práticas do PLN, para as quais também existem métodos estatísticos adequados. São apresentadas às áreas de recuperação de informações, classificação de textos e tradução automática.

Na seção final, são apresentadas as conclusões sobre este trabalho.

2 Fundamentos matemáticos

Este capítulo tem o objetivo de apresentar os conceitos básicos da teoria da probabilidade [MAN99] [MOR95] e da teoria da informação [CHA93] [EPS86] [MAN99]. O material apresentado visa possibilitar um melhor entendimento dos capítulos que seguem, cujas seções utilizam os conceitos aqui mostrados.

2.1 Teoria da probabilidade

A teoria da probabilidade trata de predizer qual a chance de que algum determinado evento aconteça. A noção de probabilidade de algum evento é formalizada através do conceito de experimento - processo pelo qual é realizada uma observação.

Em um experimento assume-se um conjunto de resultados básicos, chamado de espaço amostral, freqüentemente denotado por Ω. Espaços amostrais podem ser discretos, contendo um número infinito mas enumerável de resultados básicos, ou contínuos, contendo um número incontável de resultados básicos. Um evento é um subconjunto do espaço amostral, o resultado de um experimento. O conjunto de todos os eventos possíveis de

um espaço amostral é denominado espaço de eventos, denotado por $\mathcal{F}$, que corresponde ao conjunto de todos os subconjuntos do espaço amostral.

A probabilidade de um evento pode variar entre 0 e 1, onde 0 indica impossibilidade e 1, certeza. Uma função/distribuição de probabilidade, notada por P, distribui uma probabilidade acumulada de 1 por todo o espaço amostral. Formalmente, uma função discreta de probabilidade é qualquer função $P : \mathcal{F} \rightarrow [0, 1]$, tal que :

- $P(\Omega) = 1$
- Se $A_1, A_2, ...$ é uma seqüência de eventos mutuamente exclusivos pertencentes a $\mathcal{F}$, então

$$P(\bigcup_{j=1}^{\infty} A_j) = \sum_{j=1}^{\infty} P(A_j)$$

Chama-se $P(A)$ a probabilidade do evento A. Estes axiomas querem dizer que se determinados eventos não ocorrem simultaneamente, a probabilidade de todos ocorrerem corresponde a soma das probabilidades de cada um ocorrer.

2.1.1 Probabilidade condicional e independência

Probabilidades condicionais medem a probabilidade de eventos dado algum conhecimento.

Probabilidade a priori de um evento representa sua probabilidade antes de considerar o conhecimento adicional, e probabilidade a posteriori de um evento é a probabilidade que resulta do uso de conhecimento adicional.

A probabilidade condicional de um evento A, sabendo-se que ocorreu um evento B, é obtida por

$$P(A|B) = \frac{P(A \bigcap B)}{P(B)}$$

Se $P(B) = 0$, tem-se

$$P(A \bigcap B) = P(B)P(A|B) = P(A)P(B|A)$$

A condicionalização acima pode ser feita devido a intersecção de conjuntos ser simétrica, isto é, $A \bigcap B = B \bigcap A$. A generalização desta regra para múltiplos eventos é chamada regra em cascata[2], que será referenciada outras vezes no decorrer deste trabalho:

$$P(A_1 \bigcap ... \bigcap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \bigcap A_2)...P(A_n| \bigcap_{i=1}^{n-1} A_i)$$

Dois eventos A e B são independentes um do outro se $P(A \bigcap B) = P(A)P(B)$, o que é equivalente a dizer que $P(A) = P(A|B)$, a menos que $P(B)$ seja 0. Do contrário, os eventos são dependentes. Também pode-se dizer que dois eventos A e B são condicionalmente independentes se dado um evento C, $P(A \bigcap B|C) = P(A|C)P(B|C)$.

2.1.2 Teorema de Bayes

O teorema de Bayes nos permite trocar a ordem de dependência entre eventos, ou seja, permite calcular $P(B|A)$ a partir de $P(A|B)$.

$$P(B|A) = \frac{P(B \bigcap A)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}$$

[2]Do inglês *chain rule*.

$P(A)$ pode ser vista como uma constante de normalização. Mais genericamente, se tem-se um grupo de conjuntos B_i que particionam A, isto é, se $A \subseteq \bigcup_i B_i$ e B_i são disjuntos, então $P(A) = \sum_i P(A|B_i)P(B_i)$. Isto possibilita a geração de uma versão mais elaborada do teorema de Bayes:

$$P(B_j|A) = \frac{P(A|B_j)P(B_j)}{P(A)} = \frac{P(A|B_j)P(B_j)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}$$

2.1.3 Variáveis aleatórias

Um variável aleatória é uma função $X : \Omega \to \mathbb{R}^n$ (normalmente $n = 1$), onde $\mathbb{R}$ representa o conjunto dos números reias. Em outras palavras, variável aleatória é a função que associa a todo evento pertencente a uma partição do espaço amostral, um único número real.

Uma variável aleatória discreta é uma função $X : \Omega \to S$, onde S é um subconjunto enumerável de $\mathbb{R}$. Se $X : \Omega \to \{0, 1\}$, então X é chamado um teste de Bernoulli.

A função de probabilidade é a função que associa a cada valor x_i assumido por uma variável aleatória X, a probabilidade do evento A_i correspondente:

$$p(x_i) = P(X = x_i) = P(A_i), i = 1, 2, ..., n$$

2.1.4 Esperança e variância

A esperança consiste na média de uma variável aleatória. Se X é uma variável aleatória, sua esperança é notada por $E(X)$:

$$E(X) = \sum_{i=1}^{n} x_i p(x_i)$$

A variância de uma variável aleatória significa o grau de concentração de probabilidade em torno da média.

$$Var(X) = E((X - E(X))^2) = E(X^2) - E^2(X)$$

A variância é utilizada para medir se os valores de X tendem a ser consistentes durante os testes ou variam muito. A raiz quadrada da variância de uma variável aleatória corresponde ao desvio padrão desta.

Comumente, quando se refere a uma determinada distribuição ou conjunto de dados, a média é notada com μ, a variância como σ^2, e o desvio padrão como σ.

2.1.5 Distribuições Conjuntas e Condicionais

Mais de uma variável aleatória pode ser definida sobre um espaço amostral. Neste caso, estamos nos referindo a uma distribuição de probabilidade conjunta.

A função de probabilidade conjunta para duas variáveis aleatórias discretas X e Y é:

$$p(x, y) = P(X = x, Y = y)$$

Relacionada à função de probabilidade conjunta, há a função de probabilidade marginal, que calcula as probabilidades para os valores de cada variável separadamente:

$$p_X(x) = \sum_y p(x, y) \qquad p_Y(y) = \sum_x p(x, y)$$

Em geral, a função marginal não determina a função conjunta. Mas se X e Y são independentes, então $p(x, y) = p_X(x)p_Y(y)$.

2.1.6 Estimando funções de probabilidade

Assumir uma função de probabilidade P quando trabalha-se com linguagem não é tão simples como quando trata-se de moedas e dados. Não se conhece a probabilidade de uma determinada sentença ocorrer em um texto. Em vista disso, P deve ser estimada a partir de uma amostra de dados.

Uma importante medida para estimar P é a freqüência relativa dos resultados, que consiste na proporção de vezes que um certo resultado ocorre nos testes. Se $C(u)$ é o número de vezes que o resultado u ocorrer em N testes, então $\frac{C(u)}{N}$ é a freqüência relativa de u, notada por f_u. Se o número de testes for grande, a freqüência relativa tende a estabilizar em torno de um valor.

2.1.7 Distribuições padrão

Na prática, geralmente se encontram funções de probabilidade com as mesmas formas básicas, mas com diferentes constantes empregadas. Estas famílias de funções de probabilidade são denominadas distribuições, e as constantes que definem as diferentes funções possíveis em uma família são chamadas parâmetros.

Há distribuições discretas, como as distribuições binomial, multinomial e de Poisson; e distribuições contínuas, como as distribuições normal, exponencial e uniforme. Dentre as distribuições discretas, será explicada a distribuição binomial, e dentre as contínuas, a distribuição normal.

2.1.7.1 Distribuição binomial

Uma distribuição é dita binomial quando tem-se uma série de testes com apenas dois resultados (testes de Bernoulli), cada teste sendo independente dos demais. Por exemplo, os resultados de jogar repetidamente uma moeda seguem uma distribuição binomial. No entanto, tratando-se de um corpus de texto, não existe o caso em que a sentença seguinte é realmente independente da anterior - assim, utilizar a distribuição binomial é sempre uma aproximação. Apesar disso, para muitos propósitos, a dependência entre palavras diminui rapidamente e pode-se assumir independência. Em qualquer situação em que analisa-se a presença ou não de uma propriedade, e se está ignorando a possibilidade de dependência entre um teste e o seguinte, está implícita a utilização da distribuição binomial.

Exemplos do uso da distribuição binomial em aplicações de processamento estatístico da linguagem natural incluem percorrer um corpus em busca de uma estimativa de quantas sentenças contêm uma determinada palavra, ou descobrir quão comumente um verbo tal é usado transitivamente, procurando no corpus instâncias do verbo e anotando se seu emprego na sentença é transitivo ou não.

A família de distribuições binomiais gera o número r de sucessos em n testes, dado que a probabilidade de sucesso em qualquer teste é p:

$$b(r; n, p) = \binom{n}{r} p^r (1-p)^{n-r}$$

onde $0 \leq r \leq n$, e $\binom{n}{r} = \frac{n!}{(n-r)!r!}$, conta o número de diferentes combinações de r objetos de n, não considerando a ordem em que são escolhidos.

2.1.7.2 Distribuição normal

Muitas medidas, como as de comprimento e altura, são melhor entendidas como tendo um domínio contínuo do que discreto, e seguem uma distribuição denominada normal. Os valores das funções que seguem uma distribuição normal, funções densidade de probabilidade, não correspondem diretamente a probabilidades de um ponto do eixo x. Ao invés disso, a probabilidade de um resultado que pertence a um intervalo no eixo x é dada pela área da região delimitada pelo intervalo, o eixo x e a curva da função.

A distribuição normal tem dois parâmetros, a média μ e o desvio padrão σ, e a curva é dada por

$$n(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$$

A curva de uma distribuição normal também é chamada Gaussiana, principalmente nas comunidades de pesquisa em reconhecimento de padrões e processamento estatístico da linguagem natural.

2.1.8 Estatísticas Bayesianas

Os conceitos estatísticos vistos até o momento fazem parte da abordagem ortodoxa da estatística baseada em freqüência. No entanto, há a abordagem Bayesiana, que diverge da anterior em alguns fundamentos filosóficos.

Serão abordados a seguir alguns métodos bayesianos úteis no processamento estatístico da linguagem natural.

2.1.8.1 Atualização Bayesiana

A estatítica bayesiana mede graus de crença, e é calculada partindo-se de crenças anteriores e atualizando-as em face de uma evidência, através do uso do teorema de Bayes.

Assumindo-se que os resultados estão em seqüência e são independentes, dada uma distribuição de probabilidade a priori, pode se atualizar as crenças quando um novo dado chega, através do cálculo da máxima distribuição de probabilidade a posteriori (probabilidade MAP).

A probabilidade MAP se torna a nova probabilidade a priori e o processo se repete a cada novo dado. Este processo é denominado atualização Bayesiana.

2.1.8.2 Teoria da decisão Bayesiana

As estatísticas Bayesianas podem ser usadas para avaliar qual modelo[3] ou famílias de modelos é mais adequada para descrever certos dados.

Define-se dois diferentes modelos de probabidade de um evento e calcula-se a taxa de probabilidade, que corresponde ao quociente entre as probabilidades do evento em cada um dos modelos. Se a taxa for maior que 1, deve-se preferir o modelo cuja probabilidade foi divisor da operação, do contrário, opta-se pelo modelo do denominador, ou seja, adota-se o modelo com maior probabilidade.

2.2 Teoria da informação

O campo de teoria da informação foi desenvolvido por Claude Shannon na década de 1940. Ele estava interessado em maximizar a quantidade de informação que pode ser transmitida através de um canal de comunicação imperfeito, como uma linha telefônica com ruído. Para qualquer fonte de informação e qualquer canal de comunicação, Shannon queria determinar as taxas teóricas máximas para compressão de dados, que pode ser dada pela entropia H, e para transmissão, que é dada pela capacidade C do canal. Ele mostrou que se a informação em uma mensagem for transmitida a uma taxa mais baixa que a capacidade do canal, pode-se fazer com que a probabilidade de erros na transmisão da mensagem seja tão baixa quanto o desejado.

2.2.1 Entropia

A entropia mede a quantidade de informação em uma variável aleatória. Seja $p(x)$ a função de probabilidade de uma variável aleatória X, sobre um conjunto discreto de símbolos (ou alfabeto) $\mathcal{X}$. A entropia H é a incerteza média de uma variável aleatória.

$$H(p) = H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

A entropia também pode ser vista como o comprimento médio da mensagem necessária para transmitir o resultado de uma variável, o que é normalmente medido em bits (portanto, em logaritmo de base 2). Por

[3] Um modelo probabilístico compreende a especificação de uma distribuição e valores de parâmetros.

exemplo, se deseja-se enviar o resultado de um experimento que pode assumir 8 valores diferentes, pode-se codificar o resultado como uma mensagem de 3 bits:

1	2	3	4	5	6	7	8
001	010	011	100	101	110	111	000

O custo de transmissão de cada resultado é 3 bits. No entanto, os resultados podem ter diferentes probabilidades de ocorrência; assim, pode-se associar códigos com menos bits aos resultados mais prováveis, e códigos com mais bits para os resultados menos prováveis. Em geral, uma codificação ótima envia uma mensagem de probabilidade $p(i)$ em $\lceil -\log_2 p(i) \rceil$.

Note que: $H(X) \leq 0$; $H(X) = 0$ somente quando o valor de X é determinado, portanto não provendo uma nova informação; e a entropia cresce com o comprimento da mensagem.

2.2.2 Entropia conjunta e entropia condicional

A entropia conjunta de um par de variáveis aleatórias discretas $X, Y \sim p(x, y)$ é a quantidade média de informação necessária para especificar os valores de ambas.

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y)$$

A entropia condicional de uma variável aleatória discreta Y dada outra X, para $X, Y \sim p(x, y)$, expressa quanta informação extra em média ainda é necessária para comunicar Y dado que a outra parte conhece X.

$$H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(y|x)$$

Há também uma regra em cascata para a entropia:

$$\begin{aligned} H(X, Y) &= H(X) + H(Y|X) \\ H(X_1, ..., X_n) &= H(X_1) + H(X_2|X_1) + ... + H(X_n|X_1, ..., X_{n-1}) \end{aligned}$$

2.2.3 Informação mútua

Pela regra em cascata para entropia, tem-se $H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$. Então, $H(X) - H(X|Y) = H(Y) - H(Y|X)$. Esta diferença é chamada informação mútua entre X e Y, que consiste na redução da incerteza de uma variável aleatória devido a ter conhecimento sobre a outra, ou seja, a quantidade de informação que uma variável aleatória contém a respeito da outra.

A informação mútua I é uma medida simétrica, não negativa da informação comum em duas variáveis.

$$I(X; Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X, Y)$$

Se duas variáveis são independentes, a informação mútua é 0. Do contrário, esta cresce de cordo com entropia das variáveis. Pode-se derivar informação mútua condicional, por:

$$I(X; Y|Z) = I((X; Y)|Z) = H(X|Z) - H(X|Y, Z)$$

Uma regra em cascata para informação mútua é:

$$I(X_{1n}; Y) = I(X_1; Y) + ... + I(X_n; Y|X_1, ..., X_{n-1}) = \sum_{i=1}^{n} I(X_i; Y|X_1, ..., X_{i-1})$$

2.2.4 Modelo do canal com ruído[4]

Shannon modelou como meta da comunicação (através de uma linha telefônica) a otimização em termos de latência e precisão da transferência de mensagens na presença de ruído no canal.

A saída do canal depende probabilisticamente da entrada. Assume-se que compressão é a remoção de toda a redundância da mensagem, e a precisão na transmissão depende da insersão controlada de redundância na mensagem para que a entrada possa ser recuperada mesmo na presença de ruído. O objetivo é codificar a mensagem de tal forma que esta ocupe o mínimo espaço possível, contendo a mínima redundância necessária para a detecção e correção de erros.

O conceito central que caracteriza um canal em teoria da informação é sua capacidade C. A capacidade do canal descreve a taxa em que pode-se transmitir informação através deste com uma baixa probabilidade de não conseguir-se recuperar a entrada a partir da saída. C pode ser determinada em termos da informação mútua:

$$C = \max_{p(X)} I(X;Y)$$

De acordo com esta definição, descobre-se a capacidade do canal se for projetado um código de entrada X, cuja distribuição maximiza a informação mútua entre a entrada e a saída, considerando todas as possíveis distribuições de entrada $p(X)$. A capacidade de um canal pode ser alcançada projetando-se um código de entrada que maximiza a informação mútua entre a entrada e a saída sobre todas as possívei distribuições da entrada.

O modelo do canal com ruído é importante no processamento estatístico da linguagem natural, pois muitos problemas em PLN podem ser interpretados como uma tentativa de determinar a entrada mais parecida com uma certa saída. Uma versão simplificada deste modelo foi o centro do ressurgimento do processamento quantitativo da linguagem na década de 1970 - problemas como reconhecimento da fala e tradução automática foram mapeados para problemas de canal com ruído.

2.2.5 Entropia relativa ou divergência de Kullback-Leibler

A entropia relativa (também conhecida como divergência de Kullback-Leibler) é uma medida de quanto diferentes são duas distribuições de probabilidade (sobre o mesmo espaço de eventos).

Para duas funções de probabilidade, $p(x)$ e $q(x)$, sua entropia relativa é dada por:

$$D(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

A entropia relativa entre p e q pode ser vista como o número médio de bits que são gastos para codificar eventos de uma distribuição p, com um código baseado em uma distribuição q não muito adequada.

2.2.6 A entropia cruzada

A entropia cruzada é útil quando não se sabe a exata distribuição de probabilidade p de uma variável aleatória X que gerou determinados dados. Esta permite utilizar algum q, que é um modelo de p (uma aproximação de p).

$$H(X,q) = H(X) + D(p \parallel q)$$

A entropia pode ser entendida como o quanto surpreso fica-se por ver a próxima palavra de um texto, dada a palavra anterior já vista.

O que faz a entropia cruzada útil é que $H(X,q)$ é um limite superior da entropia $H(X)$. Para qualquer modelo q, $H(X) \leq H(X,q)$. Isto significa que pode-se usar um modelo simplificado q para ajudar a estimar

[4]Do inglês *Noisy Channel Model*.

a entropia de uma seqüência de símbolos dispostos de acordo com a probabilidade p. Quanto mais preciso for q, mais perto a entropia cruzada $H(X, q)$ estará da entropia $H(X)$. Assim, a diferença entre $H(X, q)$ e $H(X)$ é a medida da precisão do modelo.

2.2.7 A entropia da linguagem

Pode-se modelar uma linguagem usando modelos *n*-gramas ou cadeias de Markov (apresentadas na seção 5.1). Estes modelos assumem memória limitada, isto é, assume-se que a palavra seguinte depende somente das k palavras anteriores (aproximação de Markov de ordem k):

$$P(X_n = x_n | X_{n-1} = x_{n-1}, ..., X_1 = x_1) = P(X_n = x_n | X_{n-1} = x_{n-1}, ..., X_{n-k} = x_{n-k})$$

2.2.8 Perplexidade

Perplexidade pode ser vista como uma média ponderada do número de opções que uma variável aleatória pode assumir. Uma perplexidade de k significa que você está em média tão surpreso quanto você estaria se tivesse tido que escolher entre k opções equiprováveis em cada passo.

$$Perplexidade(x_{1n}, m) = 2^{H(x_{1n}, m)} = m(x_{1n}) - \frac{1}{n}$$

3 Processamento de Corpora

Este capítulo aborda questões relacionadas aos problemas encontrados quando se trabalha com textos reais no PLN. Questões importantes são, por exemplo, o pré-processamento necessário antes do trabalho com o texto, para determinar unidades como palavras e sentenças, entre outras.

De acordo com [CHU93], a disponibilização de grandes bases de textos em formato digital nos últimos anos foi a mais provável razão do ressurgimento de técnicas baseadas em conceitos estatísticos para análise de textos. Atualmente, algumas das principais organizações que distribuem corpora de texto para processamento da linguagem são: LDC - *Linguistic Data Consortium*, ELRA - *European Language Resources Association*, ICAME - *International Computer Archive of Modern English*, OTA - *Oxford Text Archive* e CHILDES - *Child Language Data Exchange System*.

Corpora consistem de dados do mundo real, ou seja, empregando linguagem como a que ocorre na fala ou escrita corrente. Assim, conforme [KRE97], as principais características observáveis de um corpus são:

- ocorrência de fenômenos lingüísticos no contexto;
- combinação de aspectos de competência e performance: os dados do corpus refletem o uso da linguagem - todos os dados, mesmo dados que são falsos em termos de competência gramatical, devem ser considerados como dados úteis;
- informações de freqüência de uma variedade de fenômentos ao invés de fenômenos selecionados;
- consideração de todos os dados existentes pelo esquema de anotação, sem distinção entre termos competentes ou não gramaticalmente.

Corpora de texto são usualmente extensos, exigindo bastantes recursos computacionais para manipulá-los. Os métodos para processamento estatístico da linguagem geralmente possuem uma etapa de coleta de freqüências a partir dos corpora. Por isso, um corpus deve ser representativo, contendo o máximo possível de estruturas existentes na linguagem pertencentes ao domínio de interesse.

Como um corpus é uma coleção especial de material textual, coletado conforme certos critérios, deve-se ser cuidadoso sobre a validade dos resultados da análise estatística obtida. Por exemplo, se um corpus foi projetado como uma amostra representativa da língua escrita, as estimativas obtidas a partir deste podem não ser adequadas à língua falada.

3.1 Tipos de corpora

Conforme apresentado em [KRE97], corpora podem ser classificados de acordo com o tipo de texto que os mesmos contêm, de acordo com o tipo de anotação que possuem, e segundo o uso a que se destinam.

3.1.1 Corpora segundo o tipo de texto

Conforme o tipo de texto que contêm, corpora podem ser classificados como balanceados, piramidais ou oportunistas, podendo haver sobreposição de tipos.

Balanceado. Consiste de diferentes gêneros de textos de tamanho ou quantidade proporcional à relevância de certo tipo de texto na linguagem em questão.

Piramidal. Apresenta desde grandes amostras de tipos de texto pouco representativos, até pequenas amostras de uma grande variedade de tipos.

Oportunista. A maioria dos corpora disponíveis pode ser caracterizada como oportunista, ou seja, contêm o que se deseja.

Paralelo. Contém textos que possuem o mesmo conteúdo, mas em duas ou mais línguas diferentes. Estes corpora são geralmente utilizados em sistemas de tradução automática.

3.1.2 Corpora segundo o tipo de anotação

De acordo com a presença ou não de anotações nos textos, ou ainda conforme o tipo de anotação existente, os corpora podem ser classificados como originais ou "crus", marcados com as categorias das palavras, *treebanks* e interpretados lingüisticamente.

Original. O texto é "tokenizado" (o processo de "tokenização"[5] é detalhado na seção 3.2.2) e limpo, isto é, caracteres de controle são eliminados. O tipo do texto, títulos e parágrafos são possivelmente marcados.

Marcado com categorias das palavras. O texto original é anotado com a categoria sintática de cada palavra.

***Treebanks*.** O texto marcado com as categoria das palavras é anotado com uma estrutura sintática. Tipicamente, uma gramática é definida, e os corpora são automaticamente analisados. As árvores de derivação são selecionadas e, se necessário, corrigidas pelos anotadores humanos. Seqüências de palavras para as quais nenhuma árvore de derivação é encontrada são omitidas ou manualmente anotadas.

Corpora interpretados lingüisticamente. Ao contrário dos *treebanks*, onde somente a categoria sintática e a estrutura da sentença são anotadas, corpora interpretados lingüisticamente visam incluir anotações de outros tipos de informação lingüística como, por exemplo, anotações semânticas.

3.1.3 Corpora segundo o uso

Corpora podem ser diferenciados conforme o uso a que se destinam: treinamento, teste ou avaliação.

[5] Do inglês *tokenization*.

Treinamento. Modelos estatísticos para processamento da linguagem natural são treinados, ou melhor, aprendem a partir de grandes corpora, tipicamente anotados.

Teste. Corpora de teste são usados para avaliar modelos estatísticos após o treinamento.

Avaliação. Corpora anotados são usados para avaliar componentes de sistemas de processamento da linguagem natural.

3.2 Condições para o processamento de corpora

Há muitas características de um texto em linguagem natural que são difíceis de serem processadas automaticamente, mesmo em baixo nível. Serão discutidos alguns problemas básicos encontrados.

3.2.1 Formatação de baixo nível

Dependendo da fonte da qual o corpus foi retirado, podem haver várias formatações e conteúdo que não são relevantes e podem ser retirados como, por exemplo, cabeçalhos de documentos e separadores, tabelas e diagramas, etc. Freqüentente, é necessária a utilização de um filtro para remover estes items dispensáveis em um documento, antes de qualquer processamento futuro do texto.

Outra questão é a opção pela diferenciação ou não entre maiúsculas e minúsculas. Pode-se optar por não diferenciar *As*, *AS* e *as*; porém, deseja-se diferenciar *oliveira* em *João Oliveira* e *oliveira* (árvore).

3.2.2 "Tokenização"

Um passo anterior ao processamento é dividir o texto de entrada em unidades chamadas *tokens*, onde cada unidade é uma palavra, ou ainda um número ou uma marca de pontuação. Este processo é chamado "tokenização".

O tratamento da pontuação tem diferentes enfoques - enquanto deseja-se manter os limites das sentenças, normalmente as marcas de pontuação internas da sentença são descartadas. No entanto, trabalhos recentes têm enfatizado a informação contida em toda pontuação - vírgulas e traços, por exemplo, podem conter informações sobre a estrutura geral do texto.

A especificação do que deve ser considerado como uma palavra é complexa; a principal informação utilizada é a ocorrência de um espaço em branco - espaço, tabulação ou o início de uma nova linha - mas mesmo este sinal não é necessariamente confiável. Os principais problemas presentes nesta questão são mostrados a seguir.

Ponto. As palavras nem sempre vêm cercadas por espaços em branco. Com freqüência, marcas de pontuação vêm junto com as palavras, como a vírgula, o ponto e vírgula, o ponto final. Torna-se problemática a remoção dos pontos finais de *tokens* que são palavras, já que um ponto pode indicar, por exemplo, uma abreviação, devendo, neste caso, fazer parte do *token*. Chama-se haplologia o fenômeno que ocorre quando uma palavra abreviada está no fim da sentença, dando ao ponto dupla função.

Apóstrofo. Uma questão difícil de ser tratada, principalmente na língua inglesa, é a presença do apóstrofo. Quando este indica, por exemplo, uma contração, como em *I'm* ou *isn't*, ou um caso possessivo, como em *boy's toy*, pode-se considerar a existência de uma ou duas palavras.

Hífen. Um hífen entre seqüências de letras indica a presença de uma ou duas palavras? Esta dúvida reflete as várias funções do hífen em um texto. O hífen pode ser usado para separar as sílabas de uma palavra, a fim de melhorar a distribuição do texto; entre as palavras que formam um substantivo composto; ou ainda, principalmente na língua inglesa, para auxiliar na qualificação de uma palavra, como em *corpus-based work*. No primeiro e no segundo caso, tende-se a considerar uma única palavra. Já no terceiro, pode-se reconhecer as palavras separadamente.

Homógrafos. São ditas homógrafas duas ou mais palavras que são escritas da mesma forma, mas têm significados diferentes. Neste casos, deve-se associar diferentes ocorrências às diferentes palavras.

Segmentação de palavras em outras línguas. Muitas línguas, como chinês e japonês, não colocam espaços entre as palavras. Nestas línguas, a tarefa de segmentação de palavras se torna muito mais importante e complexa, já que não se pode utilizar o algoritmo básico que procura por espaços em branco entre palavras. Há outras línguas que, mesmo utilizando espaço em branco entre palavras, possuem palavras compostas, como o inglês em *database* - pode-se desejar dividir tal palavra composta ou, pelo menos, ter consciência da estrutura interna da palavra.

Espaço em branco não indicando uma quebra de palavra. Contrário ao problema anterior, pode-se desejar considerar palavras separadas por um espaço como uma só palavra. Por exemplo, se *database* for considerada uma palavra única, pode-se querer tratar *data base* também como uma só palavra.

Diferentes representações de informação. Geralmente, varia de um país para outro a forma de escrever a representação de uma certa informação, como um número de telefone. Alguns países utilizam parênteses, outros traços, pontos, etc. Este é um problema comum no campo de extração de informações.

Transcritos de corpora falado. Normalmente, transcrições de corpora falados contêm contrações, representações fonéticas, fragmentos de sentenças, palavras inúteis, etc.

3.2.3 Morfologia

Outra questão refere-se ao interesse em unir ou separar as ocorrências de palavras como *comer*, *comeu* e *come*. O ato de agrupar estas formas e tratá-las como lexemas é conhecido na literatura como *stemming*, em referência ao processo de ignorar os afixos e manter apenas o radical.

Pesquisas na área de recuperação de informações mostraram que a prática de *stemming*, algumas vezes, pode melhorar bastante o desempenho de uma consulta; porém, outras vezes, o desempenho da consulta pode piorar significativamente. As principais razões para isto são três. A primeira diz respeito ao grande custo em termos de quantidade de informação necessária para o agrupamento das várias formas de um mesmo radical - por exemplo, se um usuário entra com a palavra *business* e o sistema de recuperação que usa *stemming* retornar documentos com *busy*, os resultados provavelmente não seram viáveis - ou seja, são necessárias informações específicas para determinar se certas palavras realmente devem pertencer ao mesmo grupo. A segunda razão é que, pela análise morfológica, um *token* é dividido em vários - para amenizar a explosão do vocabulário causada, geralmente é viável agrupas as informações relacionadas em blocos. A terceira está relacionada às línguas que, ao contrário do inglês, possuem sistemas complexos de inflexão e derivação, necessitando ainda mais da análise morfológica - nessa línguas, seria mais vantajoso desconsiderar a morfologia inflexional, mas não a morfologia derivacional[6].

[6] A morfologia inflexional está relacionada a inflexão das palavras em gênero, número, grau, tempo verbal,etc. Já a morfologia derivacional está ligada a palavras derivadas de outras palavras, como em *dentário* e *dentista*, que deviram de *dente*.

3.2.4 Sentenças

Pode-se pensar que "uma sentença é alguma seqüência de caracteres que termina por '.', '?' ou '!'." No entanto, como visto anteriormente, um ponto pode não indicar o fim de uma sentença, mas sim uma abreviação, ou ambas as funções ao mesmo tempo. Mesmo assim, em geral, 90% dos pontos são indicadores do fim de uma sentença. Além disso, outras marcas de pontuação podem dividir o que se considera uma sentença - freqüentemente, o que está em um ou outro ou ambos os lados de um sinal de dois pontos, ponto e vírgula ou travessão, pode ser visto como uma sentença. Porém, algumas vezes as sentenças não estão em seqüência, mas sim desordenadas. Na prática, a maioria das soluções para divisão de sentenças envolvem métodos heurísticos. No entanto, estas soluções requerem marcação manual e conhecimento do domínio por parte do desenvolvedor do sistema de "tokenização". Recentemente, pesquisas sobre métodos para automatizar o processo de delimitação de sentenças tem aumentado - há propostas de técnicas que utilizam árvores de classificação estatística, redes neuronais baseadas na distribuição das categorias das palavras, ou ainda abordagem baseadas na entropia máxima.

3.3 Treinamento e testes

Em [KRE97], são mostradas diferentes maneiras de treinar e testar modelos estatísticos para PLN.

3.3.1 Treinamento e teste com o mesmo tipo de corpus

Um corpus é dividido em duas partes que contenham tipos similares de textos. A maior parte é utilizada para treinamento, sendo denominada corpus de treinamento, e a outra é usada para teste, sendo denominada corpus de teste.

3.3.2 Treinamento e teste com tipos diferentes de corpus

Se um modelo estatístico é treinado em um tipo de corpus e testado em outro, o resultado é provavelmente pior do que o obtido no treinamento e teste em um mesmo tipo de corpus. Para conseguir-se uma idéia mais clara do desempenho do modelo, pode-se treiná-lo e testá-lo em vários subcorpora.

3.3.3 Teste com o corpus de treinamento

Testar no conjunto de treinamento significa que o mesmo corpus é usado para treinamento e teste, o que é considerado o principal equívoco na lingüística baseada em corpus. Devido ao modelo já ser otimizado no corpus de teste, o resultado do teste é muito melhor que se fosse feito com outros dados. Assim, nenhuma afirmação válida pode ser feita sobre o desempenho do modelo estatístico.

3.4 Dados marcados[7]

Embora muito possa ser feito a partir de um corpora de texto simples, sem marcação, tem-se preferido utilizar corpora onde alguns atributos da estrutura do texto são incluídos, já que, assim, pode-se obter mais informações. Esta marcação pode ser feita manualmente, automaticamente ou por ambos os métodos.

Em alguns textos são marcadas apenas estruturas básicas, como sentenças e parágrafos, enquanto em outros, são marcados vários itens, como toda a estrutura sintática. A marcação gramatical mais comum é a codificação das categorias das palavras.

[7] Do inglês *marked-up data*.

3.4.1 Esquemas de marcação

Vários esquemas têm sido usados para marcar a estrutura do texto. Primeiramente, estes foram desenvolvidos sobre uma base ad hoc. Um dos mais importantes exemplos foi o formato COCOA, usado para incluir informações de cabeçalho nos textos, como autores, data, título. A forma mais comum de marcação gramatical é indicar a categoria das palavras através da inserção de um código determinado para cada palavra. Estes códigos são normalmente indicados por algum caractere específico seguindo cada palavra, como uma barra ou um sublinhado, e então um pequeno código referente à categoria da palavra.

Atualmente, o esquema de marcação mais utilizado é o SGML (*Standard Generalized Markup Language*). SGML é uma linguagem que permite definir uma gramática para os textos, em particular para o tipo de marcação que eles contêm. A linguagem HTML é uma instância da codificação SGML. Há também a XML, um conjunto simplificado de SGML que foi particularmente desenvolvido para aplicações na World Wide Web.

3.4.2 Codificação gramatical

Normalmente, o primeiro passo da análise de um texto é realizar a marcação gramatical automática das categorias das palavras. Existem diferentes conjuntos de códigos para marcação.

Historicamente, o conjunto de códigos mais difundido tem sido o utilizado pelo corpus *American Brown* (*Brown tag set*) e as séries de conjuntos desenvolvidos na Universidade de Lancaster. Recentemente, o conjunto de códigos de marcação mais largamente utilizado computacionalmente tem sido o *Penn Treebank tag set*. Este é uma versão mais simplificada do *Brown tag set*. Em geral, conjuntos de códigos de marcação incorporam distinções morfológicas de uma língua em particular, e não podem ser diretamente aplicados a outras línguas.

O tamanho dos conjuntos de códigos varia de aproximadamente 40 a 200 diferentes códigos. Quanto maior o conjunto, mais fina a granularidade de distinção entre as palavras. Porém, os conjuntos podem escolher fazer distinções em diferentes áreas, subdividindo mais algumas áreas e menos outras. Por exemplo, o *Penn tag set* distingue 9 códigos para marcar pontuação, enquanto o *c5 tag set* tem somente 4. Presumidamente, isto indica algumas diferenças de opinião sobre o que é considerado importante.

Normalmente, um conjunto de códigos de marcação deve incluir atributos de classificação, que fornecem informações sobre a classe gramatical da palavra, e atributos preditivos, que codificam características úteis na predição do comportamento de outras palavras no contexto. Com o propósito de predição, pode-se usar códigos estritamente distribucionais, relacionados ao comportamento de palavras próximas.

4 Palavras

Neste capítulo, são abordados os principais aspectos do processamento estatístico da linguagem natural relacionados às palavras. Deste modo, apresentam-se aqui as informações e problemas que as palavras e sua distribuição no contexto representam ao PLN. Serão abordadas as seguintes questões: colocações[8], que consistem em determinadas seqüências de palavras que freqüentemente aparecem juntas, possuindo um comportamento específico; o processo de inferência estatística, que visa obter dados sobre freqüência de ocorrência das palavras e inferir sobre sua distribuição de probabilidade; redução da ambigüidade semântica daquelas palavras que possuem mais de um sentido; e aquisição léxica, que procura descobrir padrões, propriedades das palavras .

[8] Do inglês *collocations*.

4.1 *Collocations*

Nas sentenças de um corpus, uma palavra geralmente não está sozinha, mas sim rodeada de outras palavras. Uma colocação é uma expressão que consiste de duas ou mais palavras consecutivas, com um comportamento específico [MAN99]. Ou seja, as colocações de uma palavra correspondem a expressões em que esta palavra aparece em posições habituais. [ALL95] define colocações de uma palavra como "as palavras que tendem a aparecerem juntas".

De acordo com Choueka (apud [MAN99]), uma colocação tem as características de uma unidade sintática ou semântica, cujo sentido exato e não ambíguo não deriva diretamente do sentido de seus componentes. Além disso, uma expressão pode ser uma colocação mesmo que as palavras não sejam consecutivas (como no exemplo em inglês, *knock ... door*). São três as principais características de uma colocação: não composicionalidade, impossibilidade de substituição e não modificabilidade.

Uma colocação não é considerada composicional, pois seu significado não é exatamente a composição do significado de seus componentes - há uma conotação ou elemento adicional de significado que não pode ser previsto pelas partes. Por exemplo, em 'vinho branco', 'cabelo branco' e 'mulher branca', todos se referem a cores diferentes.

As palavras de um colocação não devem ser substituídas, mesmo que por outra palavra de mesmo sentido no contexto. Por exemplo, não podemos substituir 'vinho branco' por 'vinho amarelo'. Ainda, uma colocação não pode ser livremente modificada pela adição de itens lexicais ou transformações gramaticais.

As colocações podem ser divididdas em subclasses. Estas subclasses, conforme [MAN99], podem ser:

- verbos "leves": são verbos que têm pouco conteúdo semântico se utilizados sozinhos, isto é, seu complemento ajuda a definir seu significado. Por exemplo, na expressão *tomar uma decisão,* em que o verbo *tomar* não teria o mesmo significado se estivesse sozinho;
- substantivos próprios compostos: mesmo sendo diferentes das colocações léxicas, nomes próprios com mais de uma palavra são considerados colocações, pois aparecem sempre da mesma forma no texto. Por exemplo, *São Paulo*;
- expressões terminológicas: são termos que se referem a conceitos e objetos em um domínio técnico específico. Embora freqüentemente sejam termos construídos de maneira composicional, devem ser identificados como colocações, para que sejam tratados corretamente em um texto técnico. Por exemplo, *sistemas operacionais distribuídos*.

Para encontrar-se as colocações presentes em um texto, há várias abordagens: seleção das colocações por freqüência, seleção baseada em média e variância da distância entre a palavra foco e uma palavra vizinha, teste de hipótese e informação mútua. A seguir, cada uma destas abordagens é explicada.

4.1.1 Métodos para detecção de colocações

4.1.1.1 Freqüência

A maneira mais simples de encontrar colocações em um corpus é contar o número de ocorrências de diferentes seqüências de palavras. Se um grupo de palavras ocorre várias vezes, é a indicação de que este têm uma função especial. Os grupos de palavras podem ter diferentes tamanhos.

Para obter-se maior precisão na identificação de colocações, podem ser utilizadas heurísticas, como por exemplo, um filtro por categorias de palavras, que restringe os grupos resultantes aos que obedecem determinada seqüência de categorias. Por exemplo, pode-se restringir a procura por padrões de palavras em que haja um substantivo seguido se um adjetivo, como em *função linear* e *gás carbônico*.

4.1.1.2 Média e variância

Muitas colocações consistem de palavras que se relacionam de forma mais flexível, ou seja, não são adjacentes. O método da média e variância considera o padrão de variação da distância entre duas palavras, pelo número de palavras entre elas.

Calcula-se a média e a variância da distância entre duas palavras no corpus. Restringe-se a análise a uma janela de tamanho determinado ao redor da palavra atual. . A variância é calculada por:

$$s^2 = \frac{\sum_{i=1}^{n}(d_i - \overline{d})^2}{n-1}$$

onde n é o número de vezes que as duas palavras co-ocorrem, d_i é a distância entre as palavras na co-ocorrência i, e $\overline{d}$ é a média das distâncias nas ocorrências. A partir da variância, pode-se calcular o desvio padrão entre as distâncias das palavras nas diferentes ocorrências: $s = \sqrt{s^2}$.

A média e o desvio padrão caracterizam a distribuição das distãncias entre duas palavras no corpus. Pares de palavras cujo desvio padrão da distância é baixo ou nulo, indicam a existência de uma colocação.

4.1.1.3 Teste de hipótese

Como alta freqüência e baixa variância podem ser obtidas por acaso, o teste de hipótese visa determinar se a co-ocorrência de palavras é aleatória ou se realmente indica uma colocação.

Deve ser formulada uma hipótese nula H_o de que duas palavras não formam uma colocação, calcular a probabilidade *p* de que a colocação ocorrerá se H_o for verdadeira, e então rejeitar H_o se o valor de *p* for muito baixo (geralmente se o nível de significância de *p* for menor que 0,05) ou aceitar H_o.

Teste t. O teste t indica o quão provável ou improvável é a ocorrência de uma colocação. Este teste de hipótese considera a média e variância de uma amostra de medidas, e a hipótese nula diz que a amostra segue a distribuição com média μ. O teste procura a diferença entre a média observada e a esperada, corrigida pela variância dos dados, e expressa a probabilidade de se obter uma amostra com tal média e variância, assumindo que a amostra segue a distribuição normal com média μ. Calcula-se t da seguinte forma:

$$t = \frac{\overline{x} - \mu}{\sqrt{\frac{s^2}{N}}}$$

onde $\overline{x}$ é a média da amostra, s^2 é a variância da amostra, N é o tamanho da amostra e μ é a média da distribuição.

Para aplicar-se o teste t na detecção de colocações, o corpus deve ser considerado como uma seqüência de bigramas, pares de palavras.

O teste t também pode ser usado para encontrar palavras cujos padrões de co-ocorrência possibilitam a distinção entre duas palavras. Esta aplicação pode ser útil na diferenciação do significado de duas palavras. Por exemplo, pode-se encontrar palavras que melhor diferenciam os significados de *strong* e *powerful* (ambas as palavras, em português, significam forte), que são palavras usadas em situações diferentes mesmo tendo o mesmo sentido.

Teste Qui-quadrado. Quando as ocorrências em um corpus não podem ser assumidas como uma distribuição normal, pode ser utilizado outro teste de hipótese, o teste do Qui-Quadrado - χ^2, para testar a dependência entre palavras. Este compara as freqüências esperadas no caso de palavras independentes com as freqüências observadas. A estatística χ^2 soma as diferenças entre os valores observados e os esperados, dimensionados pela magnitude dos valores esperados:

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Se a diferença entre estas freqüências for grande, então pode-se rejeitar a hipótese nula de independência entre as palavras.

Um dos usos recentes do teste do qui-quadrado ocorre na identificação de pares de tradução em corpora alinhados. Ainda, uma aplicação interessante deste teste é como métrica para quantificar a similaridade entre corpora.

Teste das taxas de probabilidade. As taxas de probabilidade são mais adequadas para manipulação de dados esparsos e mais fáceis de interpretar que o teste do qui-quadrado.

Neste teste, são examinadas duas explicações alternativas para justificar a freqüência de ocorrência de um bigrama: ou a ocorrência da segunda palavra é independente da ocorrência da primeira; ou a ocorrência da segunda palavra é dependente da ocorrência da primeira.

Taxas de freqüências relativas entre dois ou mais corpora diferentes podem ser utilizadas para descobrir colocações que são características de um corpus quando comparado com outros. Desta forma, estas taxas são úteis para detecção de colocações específicas de um assunto. Para isso, pode-se comparar um texto geral com um texto de um assunto específico: aquelas palavras e expressões que ocorrem relativamente mais vezes no texto de assunto específico são provavelmente parte do vocabulário específico do domínio deste texto.

4.1.1.4 Informação mútua

A informação mútua, como visto na seção 2.2.3, é uma medida que representa "quanto uma palavra pode saber sobre a seguinte". Mais precisamente, a informação mútua pontual de uma palavra *x* sobre uma palavra *y* é a quantidade de informação provida pela ocorrência do evento representado por *y* sobre a ocorrência do evento representado por *x*.

$$I(x^{'}, y^{'}) = \log_2 \frac{P(x^{'}, y^{'})}{P(x^{'})P(y^{'})}$$

Por exemplo, a informação mútua indica que a quantidade de informação que tem-se sobre a ocorrência de *Minas* na posição i do corpus aumenta $I(Minas, Gerais)$ bits se souber-se que a palavra *Gerais* ocorre na posição $i + 1$.

4.2 Inferência estatística

A inferência estatística em geral, de acordo com [MAN99], consiste em obter dados (gerados de acordo com uma distribuição de probabilidade desconhecida) e então fazer inferências sobre sua distribuição. Um exemplo relacionado ao processamento estatístico da linguagem natural é, a partir da observação de várias instâncias de ligações de expressões preposicionais em um corpus, usar as informações sobre o comportamento destas no corpus para predizer as ligações na linguagem em geral.

Há três passos a serem considerados no processo de inferência: dividir os dados de treinamento em classes de equivalência, encontrar um bom estimador estatístico para cada classe, e combinar múltiplos estimadores.

4.2.1 Formação de classes de equivalência

A classificação dos dados de treinamento consiste em estimar uma característica com base em várias características classificatórias. Dentre estas características, pode estar, por exemplo, o conjunto das palavras anteriores a uma determinada palavra, utilizado quando deseja-se prever a palavra seguinte em um texto. Assume-se, então, que o comportamento passado é uma boa indicação sobre o que acontecerá no futuro.

Dados com valores semelhantes para determinadas características formam classes de equivalência. A classe de novos dados pode ser prevista através das classes já existentes.

As classes são definidas com base nos valores das características mais relevantes. Dividir os dados em várias classes resulta em uma boa discriminação destes. No entanto, se muitas classes forem diferenciadas, algumas podem ficar sem ou com poucos exemplares, o que não permite uma estimação confiável.

4.2.1.1 Modelos de *n*-gramas

Um exemplo de problema que necessita de inferência estatística é a previsão da palavra seguinte em uma frase, dadas as palavras anteriores. Uma seqüência de palavras pode começar de uma maneira conhecida, mas terminar por uma palavra desconhecida.

Um modo de agrupar todas as seqüências de tamanho n que começam pelas mesmas $n-1$ palavras em uma classe de equivalência é supor[9] que o contexto local prévio afeta a palavra seguinte, e contruir o modelo de Markov de ordem $(n-1)$ ou modelo de *n*-Gramas (sendo a última palavra do *n*-grama a que está sendo prevista). Os casos de *n*-gramas mais utilizados são com *n* = 2, 3 e 4, particularmente denominados bigramas, trigramas e tetragramas.

Quanto maior o valor de *n*, isto é, maior o número de classes que dividem os dados, maior a confiabilidade da inferência. No entanto, o número de parâmetros a serem estimados cresce exponencialmente em relação a *n*. Por isso, geralmente são utilizados bigramas ou trigramas em sistemas dessa natureza.

4.2.2 Estimadores estatísticos

Tendo-se os dados de treinamento já divididos em classes de equivalência, o passo seguinte é descobrir, para os dados de cada classe, como derivar uma boa estimativa de probabilidade para uma característica, com base nestes dados.

No exemplo do modelo de *n*-gramas, deseja-se conhecer a probabilidade de ocorrência do *n*-grama $w_1, ..., w_n$, notada como $P(w_1, ..., w_n)$, e prever a probabilidade de ococorrência da palavra w_n dada a ocorrência da seqüência de palavras $w_1, ..., w_{n-1}$, notada como $P(w_n|w_1, ..., w_{n-1})$.

Alguns dos métodos de estimação existentes são: *Maximum Likelihood Estimation*, leis de Laplace, Lidstone e Jeffreys-Perks, *Held Out Estimation*, *Cross-Validation* e *Good-Turing Estimation*.

4.2.2.1 *Maximum Likelihood Estimation* (MLE)

Esta estimativa é denominada "estimativa da probabilidade máxima" (MLE) porque corresponde à escolha dos valores dos parâmetros que geram a mais alta probabilidade de um evento, como um *n*-grama, no corpus de treinamento.

A MLE de um *n*-grama $w_1, ..., w_n$ é

$$P_{MLE}(w_1, ..., w_n) = \frac{C(w_1, ..., w_n)}{N}$$

onde $C(w_1, ..., w_n)$ é a freqüência do *n*-grama $w_1, ..., w_n$, e N é o número de instâncias da classe de equivalência a que o *n*-grama pertence; ou

$$P_{MLE}(w_n|w_1, ..., w_{n-1}) = \frac{C(w_1, ..., w_n)}{C(w_1, ..., w_{n-1})}$$

A MLE é geralmente inadequada ao processamento da linguagem natural, devido a os dados serem esparsos - enquanto algumas palavras são bastante comuns, a vasta maioria é pouco freqüente (e os *n*-gramas que as contêm são ainda mais incomuns). Por conseqüência, alguns eventos do corpus, seqüências de palavras, podem não estar contidos nos dados de treinamento. Para estimar a probabilidade de ocorrência destes eventos, podem ser utilizadas técnicas denominadas de *smoothing*.

[9] *Markov Assumption*

4.2.2.2 Técnicas de *smoothing*

Como algumas estruturas de palavras menos freqüentes podem não aparecer no corpus de treinamento, são necessários métodos que estimem a sua probabilidade. Estes métodos associam valores não nulos às probabilidades dos eventos não encontrados, que seria considerada zero. Para isso, diminui-se a probabilidade dos eventos encontrados, para que reste uma fatia de probabilidade para aqueles eventos não encontrados. A técnica que prevê esse desconto de probabilidades é referida como *smoothing*.

Lei de Laplace. A técnica de *smoothing* baseada na lei de Laplace, chamada informalmente de *Adding One*, tem como efeito reservar uma pequena porção do espaço de probabilidade para os eventos não conhecidos. Para isso, adiciona 1 à freqüência de cada *n*-grama encontrado no corpus de treinamento.

$$P_{LAP}(w_1, ..., w_n) = \frac{C(w_1, ..., w_2) + 1}{N + B}$$

onde $C(w_1, ..., w_n)$ é a freqüência do *n*-grama $w_1, ..., w_n$ e B é o número de classes em que as instâncias de treinamento estão divididas.

Pode-se notar que a lei de Laplace considera o tamanho do vocabulário N. Em conjuntos de dados muito esparsos, como os das aplicações de PLN, esta técnica destina muito do espaço de probabilidade para eventos não encontrados no corpus de treinamento, subestimando a probabilidade dos eventos observados.

Lei de Lidstone e Jeffrey-Perks. Já que o processo *adding one* superestima os *n*-gramas desconhecidos, pode-se adicionar um valor menor que 1, λ, à freqüência de cada *n*-grama encontrado no corpus de treinamento. Isto visa equilibrar a quantidade do espaço de probabilidade reservado aos eventos desconhecidos.

A lei de Lidstone é

$$P_{LID}(w_1, ..., w_n) = \frac{C(w_1, ..., w_n) + \lambda}{N + B\lambda}$$

onde $C(w_1, ..., w_n)$ é a freqüência do *n*-grama $w_1, ..., w_n$, B é o número de classes em que as instâncias de treinamento estão divididas, e $\lambda > 0$.

Se $\lambda = 1/2$, a lei de Lidstone é chamada *Expected Likelihood Estimation* (ELE) ou lei de Jeffreys-Perks.

4.2.2.3 Técnicas robustas

A seguir, são descritas técnicas de *smoothing* mais robustas que as mostradas anteriormente, assim tratadas por considerarem, além dos dados de treinamento, um outro conjunto de dados para validação do treinamento.

Estimação *Held Out*. Esta técnica divide os dados de treinamento em duas partes: uma para obtenção das freqüências de ocorrência dos *n*-gramas, e outra, *held out data*, também denominados dados de validação, para estimação da probabilidade de *n*-gramas desconhecidos.

Para cada *n*-grama $w_1, ..., w_n$, deve ser computada a sua freqüência nos dados de treinamento, $C_1(w_1, ..., w_n)$, e sua freqüência nos dados *held out*, $C_2(w_1, ..., w_n)$. Calcula-se N_r, que corresponde à quantidade de *n*-gramas cuja freqüência nos dados de treinamento é r. Então, calcula-se T_r, que corresponde a soma das freqüências dos *n*-gramas, cuja freqüência nos dados de treinamento é r, nos dados *held out*.

A estimativa de probabilidade de um *n*-grama é:

$$P_{ho}(w_1, ..., w_n) = \frac{T_r}{N_r N}$$

onde $r = C(w_1, ..., w_n)$.

***Cross-Validation*.** A estimação *Held Out* é viável quando há grande quantidade de dados disponíveis. Do contrário, é mais adequado utilizar cada parte dos dados ao mesmo tempo como dados de treinamento e dados de *held out*, o que é chamado de validação cruzada[10].

Em uma forma de validação cruzada bidirecional, também chamada *Deleted Estimation*, os dados de treinamento são divididos em duas partes, *a* e *b*; N_r^a é o número de *n*-gramas que ocorrem *r* vezes na parte *a* dos dados de treinamento; e T_r^{ab}, o total de ocorrências dos *n*-gramas da parte *a* na parte *b*. Uma estimação eficiente verifica as freqüências e faz *smoothing* nas duas partes dos dados, e então faz uma média ponderada das duas probabilidades, de acordo com a proporção de palavras em N_r^a e N_r^b.

$$P_{del}(w_1, ..., w_n) = \frac{T_r^{ab} + T_r^{ba}}{N(N_r^a + N_r^b)}$$

onde $r = C(w_1, ..., w_n)$.

4.2.2.4 Estimação de Good-Turing

Para determinar as freqüências e estimar as probabilidades dos eventos, a estimação de Good-Turing assume a distribuição binomial dos mesmos. Este método é adequado para um grande número de observações de dados a partir de um vasto vocabulário, e funciona bem para *n*-gramas, mesmo que as palavras de um *n*-grama não sigam uma distribuição binomial. Se $C(w_1, ..., w_n) = r > 0$,

$$P_{GT}(w_1, ..., w_n) = \frac{r^*}{N}$$

onde $r^* = ((r+1)N_{r+1})/N_r$ pode ser visto como uma freqüência ajustada.
Se $C(w_1, ..., w_n) = r = 0$,

$$P_{GT}(w_1, ..., w_n) \approx \frac{N_1}{N_0 N}$$

Uma abordagem mais simples, o Good-Turing simplificado, usa $N_r = ar^b$ (com $b < -1$) como uma curva de *smoothing*, e estima *a* e *b* por uma simples regressão linear na forma logarítmica desta equação: $\log N_r = a + b \log r$, se o valor de *r* é grande. Para valores pequenos de *r*, usa-se a medida N_r diretamente.

4.2.2.5 Similaridade

A estimação da probabilidade de eventos desconhecidos também pode ser feita através de medidas de similaridade entre palavras. Em [LEE99] é apresentado um estudo comparativo sobre diferentes funções para medir a similaridade distribucional de um bigrama. É considerada a abordagem *distance-weighted averaging*, que obtém uma estimativa para co-ocorrências desconhecidas através da combinação de estimativas para co-ocorrências envolvendo palavras similares:

$$P(w_2|w_1) = \frac{\sum_{w_3 \in S(w_1)} sim(w_1, w_3) P(w_2|w_3)}{\sum_{w_3 \in S(w_1)} sim(w_1, w_3)}$$

onde $S(w_1)$ é o conjunto de palavras similares candidatas e $sim(w_1, w_3)$ é uma função de similaridade entre w_1 e w_3.

4.2.3 Combinação de estimadores

Se há vários modelos para prever a palavra seguinte dadas as palavras anteriores, então tende-se a combiná-los para produzir um modelo ainda melhor. As combinações de métodos estimadores consideradas são: interpolação linear simples, modelo *Backing Off* de Katz e interpolação linear geral.

[10] Do inglês *cross-validation*.

4.2.3.1 Interpolação linear simples

Esta técnica visa combinar estimativas de *n*-gramas de várias ordens. Um modo de tratar o problema com dados esparsos de um modelo de trigramas é mesclar este modelo com os modelos de bigramas e unigramas (*n*-grama com *n* assumindo valor 1), que sofrem menos com a o carater esparso do corpus.

Isto pode ser feito através da interpolação linear (também denominada *finite mixture*). Quando as funções sendo interpoladas usam um subconjunto de informação de condicionamento da função mais discriminante, este método é referido como *deleted interpolation*.

$$P_{LI}(w_n|w_{n-2}, w_{n-1}) = \lambda_1 P_1(w_1) + \lambda_2 P_2(w_n|w_{n-1}) + \lambda_3 P_3(w_n|w_{n-1}, w_{n-2})$$

onde $0 \leq \lambda_i \leq 1$ e $\sum_i \lambda_i = 1$.

Os pesos, λ_i, podem ser atribuídos manualmente ou automaticamente através do algoritmo *Expectation-Maximization* (EM) .

4.2.3.2 Estratégia *Backing-off* de Katz

Em estratégias *back-off* para modelos *n*-gramas, os modelos são consultados em ordem decrescente, de acordo com a hierarquia. Ou seja, constrói-se um modelo *n*-grama baseado no modelo (*n*-1)-grama.

Se um *n*-grama de interesse apareceu mais de *k* vezes (o valor de *k* é geralmente 0 ou 1), então uma estimativa de *n*-grama é usada, mas uma quantia da estimativa MLE é descontada (reservada para *n*-gramas desconhecidos). Se um *n*-grama ocorreu *k* vezes ou menos, então deve ser usada a estimativa de um (*n*-1)-grama (probabilidade *back-off*), normalizada pela quantia de probabilidade restante e pela quantia de dados cobertos por esta estimativa.

O processo continua recursivamente.

4.2.3.3 Interpolação linear geral

Na interpolação linear simples, os pesos eram apenas número isolados, mas pode ser definido um modelo mais genérico e poderoso em que os pesos são uma função das palavras antecedentes.

Para *k* funções de probabilidade P_k, a forma geral do modelo de interpolação é

$$P_{LI}(w|h) = \sum_{i=1}^{k} \lambda_i(h) P_i(w|h)$$

onde $\forall h, 0 \leq \lambda_i(h) \leq 1$ e $\sum_i \lambda_i(h) = 1$.

4.3 Redução da ambigüidade semântica

Muitas palavras têm diferentes significados ou sentidos, sendo denominadas polissêmicas. Fora de um contexto, estas palavras são consideradas ambíguas, ou seja, há ambigüidade sobre como devem ser interpretadas. Um exemplo de palavra ambígua é a palavras *banco*, que pode ser entendida como uma instituição financeira, ou como um móvel sobre o qual se senta.

A tarefa de reduzir a ambigüidade consiste em determinar qual dos sentidos da palavra ambígua é referido em um uso particular da mesma. Para isto, é observado o contexto de uso da palavra.

Assume-se que uma palavra tem um número finito de sentidos, freqüentemente apresentados por um dicionário, thesaurus, ou outra fonte de referência. A função de um programa que visa identificar o sentido usado de uma palavra ambígua é fazer uma escolha forçada entre os diferentes sentidos desta palavra, baseando-se no contexto de uso.

A resolução da ambigüidade de sentido das palavras tem grande importância em muitas aplicações de PLN, como sistemas de tradução automática, em que a palavra correspondente na outra língua pode variar conforme

o sentido da palavra na língua atual do documento, e em sistemas para recuperação de informações, que devem retornar apenas documentos que atendam a determinado sentido de uma palavra ambígua, expressado de alguma forma na consulta.

Há também outro tipo de ambigüidade, em que uma palavra pode ser usada em diferentes categorias morfo-sintáticas. Por exemplo, a palavra *canto*, que é um substantivo quando significa o canto de um objeto ou lugar, ou é a flexão na primeira pessoa do singular do presente do indicativo do verbo *cantar*. Determinar a categoria de uma palavra de acordo com seu uso é uma tarefa denominada marcação.

A natureza da ambigüidade e a técnica utilizada para seu tratamento variam de acordo com o material que está disponível para treinamento do sistema de redução da ambigüidade de sentido. Nas subseções seguintes, após uma discussão inicial sobre metodologia, serão abordados os diferentes tipos de técnicas de redução da ambigüidade, classificadas conforme o material de treinamento. O tratamento da ambigüidade é dito supervisionado, quando utiliza-se dados de treinamento marcados; é dito baseado em dicionário, quando utiliza-se recursos léxicos como dicionários e thesauri; e é dito não supervisionado, quando utiliza-se corpora de textos não marcados.

4.3.1 Metodologia

Há muitas questões metodológicas importantes no contexto de redução da ambiguidade de sentido, como aprendizado supervisionado versus não supervisionado e a utilização de dados de avaliação artificiais, conhecidos como pseuso-palavras.

4.3.1.1 Aprendizado supervisionado e não supervisionado

No aprendizado supervisionado, o rótulo de uma palavra é conhecido. Já no aprendizado não supervisionado, não se sabe a classificação dos dados. Desta forma, o aprendizado supervisionado pode ser visto como uma tarefa de classificação, enquanto o não supervisionado pode ser visto como uma tarefa de agrupamento[11].

Devido ao fato de a produção de dados de treinamento marcados ser custosa, deseja-se freqüentemente aprender a partir de dados não marcados. Isto implica adicionar aos algoritmos o uso de várias fontes de conhecimento, como dicionários ou estruturas de dados mais complexas, como textos bilíngües alinhados. Ainda, existem métodos que são inicializados com dados de treinamento marcados, mas estes dados são expandidos por um aprendizado posterior com dados não marcados.

4.3.1.2 Pseudo-palavras

Para testar-se a performance dos algoritmos para tratamento da ambigüidade de palavras, necessita-se primeiramente resolver de modo manual a ambigüidade de um grande número de ocorrências, o que é uma tarefa demorada e árdua. Nestes casos, em que os dados disponíveis são trabalhosos para serem manipulados, é mais conveniente, conforme [MAN99], gerar dados artificiais de avaliação para a comparação e melhora dos algoritmos para processamento dos textos. Estes dados artificiais são denominados “pseudo-palavras”.

4.3.2 Redução supervisionada da ambigüidade

Em uma abordagem supervisionada para redução da ambigüidade, é utilizado para treinamento um corpus em que as palavras ambíguas já foram tratadas. Há um conjunto de treinamento composto por amostras em que cada ocorrência de uma palavra ambígua é anotada com um rótulo semântico. A tarefa é construir um classificador que classifique corretamente novos casos, baseados no contexto de uso.

A seguir, serão apresentados dois exemplos de algoritmos supervisionados, que fazem parte de uma das duas importantes abordagens teóricas do processamento estatístico da linguagem: classificação bayesiana e teoria da informação.

[11] Em inglês, o termo utilizado é *clustering*.

4.3.2.1 Classificação bayesiana

A idéia do classificador bayesiano, proposto por Gale et al. (apud [MAN99]), é procurar pelas palavras $v_1, ..., v_j, ..., v_J$ que cercam a palavra ambígua w em uma determinada janela no contexto c. Cada palavra contribui com informações sobre o sentido s_k da palavra ambígua. Ou seja, o classificador não seleciona determinadas características, mas sim combina todas as características evidentes. O treinamento supervisionado do classificador assume a existência de um corpus onde cada ocorrência da palavra ambígua é rotulada com o sentido correto.

O classificador de Bayes aplica a regra de decisão de Bayes para escolher um sentido. A regra é a seguinte: decide s' se $P(s'|c) > P(s_k|c)$, para $s_k \neq s'$. A regra de decisão de Bayes minimiza a probabilidade de erro - para cada caso, escolhe o sentido com maior probabilidade condicional.

Usualmente, não se sabe o valor de $P(s_k|c)$, mas pode-se computá-lo através da regra de Bayes:

$$P(s_k|c) = \frac{P(c|s_k)}{P(c)} P(s_k)$$

$P(s_k)$ é a probabilidade a priori do sentido s_k, a probabilidade de a palavra ambígua w ter o sentido s_k se nada se conhece sobre o contexto.

Desta forma, associa-se a w o sentido s_k quando $s' = \arg\max_{s_k} P(s_k|c)$.

O classificador de Gale et al. é uma instância de um tipo particular de classificador de Bayes, o classificador *Naive Bayes*, que é eficiente na tarefa de combinar evidências de um grande número de características. Neste caso, as palavras v_j que descrevem o contexto de w são as características disponíveis. A hipótese de *Naive Bayes* que os atributos utilizados para descrição são todos condicionalmente dependentes:

$$P(c|s_k) = P(\{v_j|v_j \; in \; c\}|s_k) = \prod_{v_j \; em \; c} P(v_j|s_k)$$

Pela hipótese de *Naive Bayes*, a estrutura e ordem das palavras no contexto são ignoradas, formando um "saco de palavras"[12], e a presença de uma palavras no saco é independente da outra.

Com a hipótese de *Naive Bayes*, tem-se uma regra de decisão modificada para classificação: escolhe s' se $s' = \arg\max_{s_k} [\log P(s_k) + \sum_{v_j \; em \; c} \log P(v_j|s_k)]$. $P(v_j|s_k)$ e $P(s_k)$ são computadas pela estimação da probabilidade máxima a partir do corpus de treinamento marcado:

$$P(v_j|s_k) = \frac{C(v_j, s_k)}{C(s_k)} \qquad e \qquad P(s_k) = \frac{C(s_k)}{C(w)}$$

onde $C(v_j, s_k)$ é o número de ocorrências de v_j em um contexto de sentido s_k no corpus de treinamento, $C(s_k)$ é o número de ocorrências de s_k no corpus de treinamento, e $C(w)$ é o número toal de ocorrências da palavra ambígua w.

4.3.2.2 Teoria da informação

O algoritmo baseado na teoria da informação, proposto por Brown et al. (apud [MAN99]), procura por uma única caracterísctca contextual que indique com segurança o sentido da palavra ambígua que está sendo usado. Estas características podem ser sensíveis à estrutura do texto.

Um exemplo de característica utilizada é o objeto, que pode determinar o sentido de um verbo - se o objeto for x_1, o sentido do verbo será X_1, se for x_2, o sentido será X_2. Para o bom uso de uma característica, seus valores devem ser categorizados conforme o sentido que indicam, ou seja, x_1 indica X_1, x_2 indica X_2. Os valores x_i que indicam o sentido X_i, são agrupados no conjunto Q_i. A partir disto, a tarefa de resolução da ambigüidade consiste em:

[12] A expressão "saco de palavras" é utilizada devido a distribuição das palavras na sentença não ser considerada.

1. Para a ocorrência da palavra ambígua, determinar o valor x_i da característica;

2. Se o valor é x_i está contido em Q_1, associa-se a ocorrêcia ao sentido X_1; se o valor x_i está contido em Q_2, associa-se a ocorrência ao sentido X_2; e assim por diante.

4.3.3 Redução da ambigüidade baseada em dicionário

Se não se tem informações sobre a categorização do sentido de instâncias específicas de uma palavra, pode-se optar por uma caracterização geral dos sentidos. Para isso, deve-se confiar nas definições de sentidos contidas em dicionários e thesauri.

Três tipos de infomação têm sido usados: definições de sentido obtidas diretamente do dicionário, categorias semânticas das palavras (obtidas a partir de um thesaurus), e traduções dos diferentes sentidos, extraídas de um dicionário bilíngüe, cujas distribuições no corpus de língua estrangeira são analisadas para resolução da ambigüidade.

Além disso, uma investigação das propriedades distribucionais dos sentidos pode levar a melhoras significativas no sistema de tratamento da ambigüidade, pois, comumente, palavras ambíguas são usadas com um único sentido em um dado discurso e em uma dada colocação.

4.3.3.1 Redução da ambigüidade baseada nas definições dos sentidos

Lesk, autor desta técnica, baseou-se na simples idéia de que as definições das palavras no dicionário são provavelmente bons indicadores dos sentidos que elas definem.

Sejam $D_1, ..., D_k$ as diferentes definições dos sentidos $s_1, ..., s_K$ da palavra ambígua w. Seja v_j uma palavra que ocorre no contexto c em que w está inserida, e E_{v_j} o conjunto das definições de v_j encontradas no dicionário (se v_j tem vários sentidos $s_{j_1}, ..., s_{j_L}$, então $E_{v_j} = \bigcup_{j_i} D_{j_i}$). O sentido em uso s_k de w é determinado pelo número de palavras comuns em D_k e em $\bigcup_{v_j \, em \, c} E_{v_j}$ (união dos conjuntos de definições de todas as palavras do contexto c).

Em [MIH99], é apresentado um método que visa resolver a ambigüidade de substantivos, verbos, advérbios e adjetivos em corpus de textos sem restrições, utilizando os sentidos providos na WordNet[13]. Neste método, os sentidos das palavras são ordenados, primeiramente, de acordo com estatísticas para co-ocorrências de pares de palavras, obtidas na Internet através do *site* de busca Altavista[14]. Em seguida, a ordenação é refinada através de um método diferente, denominado densidade semântica, utilizando-se a WordNet. O método de densidade semântica se propõe a medir a relação entre as palavras, utilizando informações que cada conceito possui na WordNet, que atuam como um micro-contexto para cada conceito. As palavras são analizadas em pares, para que a ambigüidade de uma seja resolvida no contexto da outra, mas o trabalho também analisa a possibilidade de tuplas de mais que duas palavras serem utilizadas.

4.3.3.2 Redução da ambigüidade baseada em thesaurus

A principal idéia desta técnica é que as categorias semânticas das palavras no contexto determinam a categoria semântica do contexto como um todo. A categoria do contexto, então, passa a determinar quais sentidos são usados para as palavras .

Walker (apud [MAN99]) propôs um algoritmo simples baseado em thesaurus: a cada palavra são associados um ou mais códigos que correspondem a seus diferentes significados. Se vários códigos são associados a uma mesma palavra, assume-se que estes correspondem a diferentes sentidos da mesma. Seja $t(s_k)$ o código do sentido s_k da palavra ambígua w. A ambigüidade de w pode ser resolvida através da contagem do número de

[13] A WordNet é uma grande base de dados eletrônica que contém as relações léxicas entre as palavras da língua inglesa. A WordNet é dividida em três bases de dados: uma para substantivos, outra para verbos, e uma terceira para adjetivos e advérbios. Cada base de dados consiste de um conjunto de itens léxicos que correspondem a formas ortográficas únicas, acompanhadas pelo conjunto de significados associados a cada forma.

[14] Disponível em http://www.altavista.com

palavras para as quais o thesaurus contém $t(s_k)$ como possível tema. Então, escolhe-se o sentido com maior número de ocorrências.

O problema deste algoritmo é que uma categorização geral de palavras em temas é freqüentemente difícil de ser obtida para um domínio específico.

Yarowski (apud [MAN99]) propôs outro algoritmo para adaptação da classificação por temas a um corpus. Este algoritmo adiciona palavras a uma categoria t_i se estas ocorrem um número de vezes maior que um limite α no contexto de t_i no corpus.

4.3.3.3 Redução da ambigüidade baseada em traduções de um corpus em outra língua

O foco desta técnica, proposta por Dagan et al. (apud [MAN99]), é que a ambigüidade das palavras pode ser resolvida através da observação de como as palavras são traduzidas em outras línguas.

Por exemplo, a palavra *interest* tem mais de um sentido em inglês, com diferentes traduções em alemão: *Beteiligung*, se significar participação nas cotas de uma empresa, e *Interesse*, se significar atenção, preocupação. Para resolver a ambigüidade de uma ocorrência de *interest* em inglês, identifica-se a expressão em que esta ocorre e procura-se no corpus em alemão por instâncias da mesma expressão; associa-se o sentido correspondente toda vez que *interest* é usado na expressão identificada.

4.3.3.4 Um sentido por discurso, um sentido por colocação

Há restrições entre as diferentes ocorrências de uma palavra ambígua em um corpus que podem ser exploradas para resolução da ambigüidade. Yarowsky enfatizou duas destas restrições:

- ◇ "um sentido por discurso": o sentido da palavra é altamente consistente em um dado documento;
- ◇ "um sentido por colocação": palavras próximas provêem informações sobre o sentido da palavra ambígua, dependendo da distância relativa, ordem e relação sintática.

4.3.4 Redução não supervisionada da ambigüidade

Há situações em que nem mesmo uma pequena quantidade de informação léxica ou semântica sobre as palavras de um corpus está disponível. O princípio da redução não supervisionada da ambigüidade é interpretar os sentidos das palavras sem ter recursos como textos marcados, dicionários ou thesauri. Por isso, não é possível realizar marcação semântica no corpus, pois seria necessário rotular as ocorrências como pertencentes a um sentido ou outro (a marcação semântica requer que sejam fornecidas caracterizações dos sentidos).

No entanto, pode-se realizar a discriminação dos sentidos. Esta discriminação é realizada separando-se os contextos de uma palavra ambígua em um número de grupos, e os grupos são discriminados, sem serem marcados. Muitos algoritmos para discriminação de sentidos têm sido propostos. Um deste algoritmos, proposto por Schütze, denominado discriminação de grupos de contexto[15], é um algoritmo completamente não supervisionado que agrupa ocorrências não rotuladas.

Este algoritmo utiliza o mesmo modelo probabilístico utilizado por Gale et al., mostrado na seção 4.3.2.1. Para uma palavra ambígua w com sentidos $s_1, ..., s_k, ...s_K$, estima-se a probabilidade condicional de cada palavra v_j (contida em uma janela que cerca w) ocorrer no contexto em que w possui o sentido s_k, isto é, $P(v_j|s_k)$. No entanto, ao contrário do classificador bayesiano de Gale et al., os parâmetros $P(v_j|s_k)$ são inicializados aleatóriamente, ao invés de serem obtidos a partir de um conjunto de dados de treinamento marcados. $P(v_j|s_k)$ é reestimada (o processo de reestimação de parâmetros é descrito na seção 5.1.2.3) pelo algoritmo de maximização da expectativa. Então, computa-se, para cada contexto c_i de w, a probabilidade $P(c_i|s_k)$. Esta categorização preliminar dos dados pode ser usada como dados de treinamento. Assim, reestima-se os parâmetros $P(v_j|s_k)$ para maximizar a probabilidade dos dados dado o modelo. O critério de parada do algoritmo é a situação em a probabilidade não é mais incrementada significativamente.

[15] Do inglês *context-group discrimination*.

Depois de estimados os parâmetros do modelo, $P(v_j|s_k)$, pode-se resolver a ambigüidade do contexto de w através do cálculo da probabilidade de cada um dos sentidos, baseado nas palavras v_j que ocorrem no contexto. Então, utiliza-se a hipótese de *Naive Bayes* e aplica-se a regra de decisão de Bayes, mostradas também na seção 4.3.2.1.

4.4 Aquisição léxica

Aquisição léxica consiste em obter-se propriedades sintáticas e semânticas das palavras. O objetivo geral da aquisição léxica, conforme [MAN99], é desenvolver algoritmos e técnicas estatísticas para suprir as falhas dos dicionários existentes, através da procura por padrões de ocorrência das palavras em corpora extensos.

A maioria das propriedades das palavras que são de interesse em PLN não são fornecidas completamente em dicionários existentes. Isto ocorre devido ao dinamismo da linguagem natural - constantemente, novas palavras e novos usos de palavras já existentes são inventados - e também ao fato de os dicionários serem desenvolvidos de acordo com as necessidades dos humanos que, em geral, não necessitam de informações quantitativas em relação às palavras. A aquisição léxica é capaz de suprir esta deficiência dos dicionários, provendo as informações quantitativas, além de informações qualitativas.

São vários os problemas relacionados à aquisição léxica. A aquisição de colocações e a resolução da ambigüidade semântica, vistos nas seções 4.1 e 4.3 respectivamente, são problemas de aquisição léxica, apresentados separadamente devido a sua importância no PLN. A seguir, serão vistos outros problemas de aquisição léxica: subcategorização de verbos, que consiste em classificar os verbos de acordo com os tipos de complementos que permitem; ambigüidade de ligação, que consiste em determinar a que elemento da sentença está ligada uma expressão ambígüa; preferências selecionais, que consistem nas restrições semânticas que um lexema impõe a seus complementos; e similaridade semântica, que consiste na determinação do significado de uma palavra nova a partir de uma medida de similaridade desta palavra em relação às palavras já conhecidas.

4.4.1 Subcategorização de verbos

Os verbos podem ser classificados de acordo com os tipos de complemento que admitem. Este processo é denominado subcategorização. Diz-se que um verbo é subcategorizado por um complemento particular, que pode ser o sujeito da oração, o objeto ou uma expressão preposicional, entre outros. O conjunto de categorias sintáticas que podem aparecer junto com o verbo em uma sentença é denominado quadro de subcategorização[16]. Alguns dos quadros de subcategorização mais comuns são mostrados na figura 1.

Quadro	Verbos	Exemplo
sujeito	intransitivos	A mulher caminhava.
sujeito e objeto direto	transitivos diretos	Ele ama Ana.
sujeito e expressão preposicional	transitivos indiretos	Maria telefonou para sua mãe.
sujeito, objeto direto e expressão preposicional	transitivos diretos e indiretos	Carla colocou o prato na mesa.
sujeito e oração complementar		Eu sei que ela mora em Paris.
sujeito e infinitivo		Ela deseja comer.

Figura 1: Exemplos de quadros de subcategorização.

Pode-se ver cada classe como um grupo de verbos com um determinado conjunto de argumentos semânticos. Assim, cada subclasse das classes expressa estes argumentos semânticos através de diferentes funções sintáticas.

[16]Do inglês *subcategorization frame*.

Saber o quadro de subcategorização dos verbos é importante para a análise sintática de uma sentença, para que se possa associar corretamente os complementos aos verbos ou a outros elementos da sentença.

Um algoritmo para aprendizado de quadros de subcategorização foi proposto por Brent (apud [MAN99]). Este algoritmo decide, baseado no corpus, se os complementos de um verbo v se excaixam no quatro q. Esta decisão é feita em dois passos:

1. Define-se um padrão regular de palavras e categorias sintáticas que indiquem a presença de q com alto grau de certeza. Para uma determinada suposição de padrão c^j, define-se a probabilidade de erro ϵ_j, que indica o quão equivocado é o fato de associar-se o quadro q ao verbo v, com base no padrão suposto c^j;

2. Assume-se que o quadro não é apropriado para o verbo - hipótese nula H_0. Rejeita-se H_0 se c^j indicar com alta probabilidade que esta hipótese está errada. Uma vez que os padrões dos quadros de interesse tenham sido definidos, pode-se analisar o corpus e contar, para cada combinação verbo-quadro, o número de vezes que o padrão para o quadro ocorre com o verbo. Supondo que o verbo v^i ocorra um total de n vezes no corpus, e que há $m \leq n$ ocorrências de um padrão para o quadro q^i, pode-se rejeitar a hipótese nula H_0 de que v^i não permite q^j com a p_E probabilidade de erro:

$$p_E = P(v^i(f^j) = 0 | C(v^i, c^i) \geq m) = \sum_{r=m}^{n} \binom{n}{r} \epsilon_j^r (1 - \epsilon_j)^{n-r}$$

   onde $v^i(f^i) = 0$ significa que o verbo v^i não admite o quadro f^j, $C(v^i, c^j)$ é o número de vezes que v^i ocorre com o padrão c^j, e ϵ_j é a taxa de erro para o padrão f^j, ou seja, é a probabilidade de encontrar-se o padrão c^j para uma ocorrência particular do verbo, embora o quadro não esteja realmente sendo usado. p_E é a probabilidade de ocorrência dos dados observados, dado que H_0 seja correta.

O algoritmo de Brent é bastante eficiente na associação de quadros aos verbos - aproximadamente 100% de associações corretas. No entanto, este algoritmo não é tão eficiente na seleção dos quadros. A porcentagem de acerto na seleção é ainda menor quando se trata de verbos pouco freqüentes.

Manning (apud [MAN99]) trata deste problema utilizando um sistema de marcação e realizando a detecção de padrões sobre o resultado da marcação. O método de Manning pode aprender um grande número de quadros de subcategorização, mesmo que utilize padrões pouco confiáveis. Mas os resultados deste método podem ser ainda melhorados através da incorporação de conhecimento prévio sobre os quadros de subcategorização dos verbos.

4.4.2 Ambigüidade de ligação

Na análise sintática de uma seqüência de palavras, é gerada uma árvore de derivação desta seqüência de acordo com uma gramática. Se puderem ser geradas diferentes árvores de derivação para uma mesma seqüência de palavras, isto significa a ocorrência do fenômeno denominado ambigüidade sintática.

Um tipo de ambigüidade sintática particularmente freqüente é a ambigüidade de ligação, que ocorre com expressões que podem ser ligadas a mais de um nodo na árvore de derivação. Este é um problema de natureza sintática, mas pode ser resolvido através de propriedades léxicas das palavras.

A ambigüidade de ligação pode estar relacionada a expressões e orações adverbiais, expressões preposicionais e substantivos compostos por mais de duas palavras.

Porém, a ligação de sintagmas preposicionais constitui o problema de ambigüidade de ligação mais discutido na literatura de processamento estatístico da linguagem. Um exemplo de sentença ambígua em relação à ligação de uma expressão preposicional é: *A criança comeu o bolo com a colher.* A expressão *com a colher* pode estar ligada a *comeu*, com o sentido de que a criança comeu o bolo utilizando uma colher, ou a expressão pode estar ligada a *bolo*, com o sentido de que a criança escolheu, dentre vários bolos, comer o que tinha uma colher.

Na maioria dos casos, a ligação correta pode ser determinada por estatísticas lexicais simples, como número de co-ocorrências entre verbo v e preposição p, e o número de co-ocorrências entre o substantivo s e a preposição p.

Um modelo simples, baseado nestas informações, computa a taxa de probabilidade λ:

$$\lambda(v, s, p) = \log \frac{P(p|v)}{P(p|s)}$$

onde $P(p|v)$ é a probabilidade de encontrar-se uma expressão preposicional que contém p após o verbo v, e $P(p|s)$ é a probabilidade de encontrar-se uma expressão preposicional que contém p após o substantivo s. Liga-se a expressão preposicional ao verbo se $\lambda(v, s, p) > 0$, e ao susbstantivo se $\lambda(v, s, p) < 0$.

O problema deste método está em ignorar a tendência por operações locais, isto é, o fato de que deve-se ligar a expressão ambígua ao nodo mais inferior na árvore quando as demais informações não forem suficientes para determinar a ligação correta. Esta tendência deve-se ao fato de o nodo mais inferior estar mais "recente na memória", pois aparece por último na sentença. No caso da ligação de uma expressão preposicional ambígua, no nodo mais inferior está o substantivo.

Esta tendência a optar-se pelo substantivo mais inferior é formalizada pelo método de Hindle e Rooth para determinar a ligação de expressões preposicionais, com base em informações lexicais.

4.4.2.1 Método de Hindle e Rooth

O espaço de estados é definido como sendo o conjunto de todas as orações que têm um verbo transitivo, um substantivo seguindo o verbo, e uma expressão preposicional seguindo o substantivo.

Estima-se o quão provável é a ligação da preposição, contida na expressão preposicional, ao verbo ou ao substantivo. Sejam VA_p e SA_p duas variáveis aleatórias. Se há uma expressão preposicional iniciada pela preposição p, seguindo o verbo v, que se ligue a v, então $VA_p = 1$, senão $VA_p = 0$. Se há uma expressão preposicional iniciada pela preposição p, seguindo o substantivo s, que se ligue a s, então $SA_p = 1$, senão $SA_p = 0$.

De acordo com [CHA93], Hindle e Rooth assumem que:

$$P(\text{ligacao}(p) = s|v, s) > P(\text{ligacao}(p) = v|v, s) \Leftrightarrow P(SA_p = 1|s) > P(VA_p = 1|v)$$

$P(VA_p = 1|v)$ e $P(SA_p = 1|s)$ podem ser estimados pelo cálculo da probabilidade máxima:

$$P(VA_p = 1|v) = \frac{C(v, p)}{C(v)} \qquad e \qquad P(SA_p = 1|s) = \frac{C(s, p)}{C(s)}$$

onde $C(v)$ e $C(s)$ são, respectivamente, o número de ocorrências de v e s no corpus, e $C(v, p)$ e $C(s, p)$ são, respectivamente, o número de vezes que p está ligada a v e que p está ligada a s.

Como não se sabe a que nodo a expressão preposicional está ligada, não se pode obter estatísticas. Então, Hindle e Rooth utilizam a seguinte heurística para determinar $C(v, p)$ e $C(s, p)$:

1. Se um substantivo é seguido por uma expressão preposicional, mas não há verbo precedente, incrementa-se $C(s, p)$. Isto acontece, por exemplo, se o substantivo é o sujeito da frase;
2. Se um verbo na voz passiva é seguido por uma expressão preposicional, incrementa-se $C(v, p)$. Por exemplo, em *O cachorro foi machucado na perna*;
3. Se uma expressão preposicional segue um sintagma nominal e um verbo, mas o sintagma nominal é um pronome, incrementa-se $C(v, p)$. Por exemplo, em *Catatina viu o garoto no parque*;
4. Se uma expressão preposicional segue um substantivo e um verbo, observa-se se as probabilidades baseadas nas decisões de 1 a 3 favorecem uma ou outra ligação. Incrementa-se o contador da ligação favorecida.
5. Do contrário, incrementa-se ambos $C(v, p)$ e $C(s, p)$ com 0.5.

4.4.3 Preferências selecionais

A maioria dos verbos prefere argumentos de um determinado tipo. Tais regularidades são denominadas preferências ou restrições selecionais. Por exemplo, os objetos do verbo *comer* tendem a ser alimentos, os sujeitos do verbo *latir* tendem a ser cães.

Estas restrições semânticas dos argumentos de um verbo são análogas às restrições sintáticas vistas anteriormente na seção 4.4.1 sobre subcategorização de verbos em objetos, sujeitos, infinitivos, etc.

A aquisição de preferências selecionais pode ser utilizada para inferir-se parte do sentido de uma palavra ocorrida em um texto, mas não encontrada no dicionário. Outro uso importante de preferências selecionais é a elaboração de um *ranking* dos possíveis resultados da análise sintática de uma sentença - dá-se maiores pesos aos resultados em que o verbo tem os argumentos que prefere, o que permite escolher entre análises igualmente adequadas do ponto de vista sintático.

Resnik (apud [MAN99]) propôs um modelo para aquisição de preferências selecionais, que pode ser aplicado a qualquer classe de palavras que imponha restrições semânticas. No entanto, será considerado apenas o caso da relação entre verbos e objetos diretos. O modelo formaliza as preferências selecionais, utilizando as noções de intensidade de preferência selecional e associação selecional.

A intensidade da preferência selecional mede a "força" com que o verbo restringe seu objeto direto. Esta medida é definida pela divergência de Kullback-Leibler (vista em 2.2.5) entre a distribuição de objetos diretos para verbos em geral e a distribuição de objetos diretos do verbo que se está tentando caracterizar. Assim, tem-se:

$$I(v) = D(P(C|v) \parallel P(C) = \sum_c P(c|v) \log \frac{P(c|v)}{P(c)}$$

onde $P(c)$ é a distribuição de probabilidade da classes dos substantivos que estão nos núcleos dos objetos diretos em geral, e $P(c|v)$ é a distribuição de probabilidade da classes dos substantivos que estão nos núcleos dos objetos diretos do verbo v.

Com base na intensidade da preferência selecional, pode-se definir a associação selecional entre um verbo v e uma classe de substantivos c, como segue:

$$A(v,c) = \frac{P(c|v) \log \frac{P(c|v)}{P(c)}}{I(v)}$$

Necessita-se, então, de uma regra para relacionar a medida de associação a substantivos. Se o substantivo s não está na classe c, define-se $A(v,s) \stackrel{def}{=} A(v,c)$. Se o substantivo é um membro de várias classes, então define-se sua medida de associação como a maior associação dentre as de suas classes.

Para estimar-se a probabilidade de ocorrência do objeto direto, cujo núcleo é um substantivo da classe c, dado que o verbo v ocorra, tem-se:

$$P(c|v) = \frac{P(v,c)}{P(v)}$$

A probabilidade máxima estimada para $P(v)$ é $\frac{C(v)}{\sum_{v'} C(v')}$, que corresponde à freqüência relativa de v em relação a todos os verbos. E $P(v,c)$ pode ser estimado, de acordo com Resnik, com

$$P(v,c) = \frac{1}{N} \sum_{s \in \text{palavras}(c)} \frac{1}{|\text{classes}(s)|} C(v,s)$$

onde N é o número total de pares verbo-objeto no corpus, $\text{palavras}(c)$ é o conjunto de todos os substantivos da classe c, $\text{classes}(s)$ é o número de classes de substantivos que contêm s, e $C(v,s)$ é o número de pares verbo-objeto em que o verbo é v e o núcleo do objeto é s. Se um substantivo que é membro de duas classes c_1 e c_2 ocorre com v, então atribui-se metade da ocorrência para $P(v,c_1)$ e metade para $P(v,c_2)$.

4.4.4 Similaridade semântica

A maioria dos esforços para aquisição de propriedades semânticas das palavras têm sido focados na similaridade semântica. Isto é, a determinação do significado de uma palavra nova é realizada a partir da obtenção automática de uma medida de similaridade desta palavra em relação às palavras já conhecidas.

A medida de similaridade semântica também é utilizada para generalização, já que supõe-se que palavras similares se comportem de forma semelhante. Por exemplo, no problema de preferências selecionais (visto na seção anterior), não se tendo conhecimento sobre uma palavra, mas sabendo-se que esta é semanticamente similar a *maçã* e *banana*, pressupõe-se que a palavra desconhecida também seja aceita pelo verbo *comer*.

Em sistemas de recuperação de informações, a similaridade semântica pode ser usada para expandir uma consulta especificada por uma palavra-chave, procurando também por documentos que contenham palavras similares à palavra fornecida na consulta.

A noção de similaridade semântica parece intuitiva; porém, seu entendimento pode variar. Por alguns, a similaridade semântica é vista como uma extensão de sinonímia[17]. Algumas vezes, a similaridade semântica indica que duas palavras pertencem a um mesmo domínio semântico. Partindo-se deste entendimento do termo, palavras são similares se referem-se a entidades que provavelmente ocorrem juntas, como *médico*, *enfermeira*, *febril*, *cancerígeno*. Tais palavras podem se referir a entidades completamente diferentes, mesmo de diferentes categorias sintáticas. Miller e Charles (apud [MAN99]) tentaram clarear a noção de similaridade semântica, definindo-a como o grau em que uma palavra pode ser substituída por outra no contexto.

Para medir-se a similaridade semântica das palavras, podem ser utilizadas medidas probabilísticas, onde a similaridade semântica entre duas palavras é tratada como a similaridade entre duas distribuições de probabilidade.

Para obter-se as medidas probabilísticas, são construídas matrizes bidimensionais que fornecem diferentes tipos de similaridade entre as palavras, conforme forem os dados representados em cada dimensão. A seguir, são descritas três matrizes diferentes: matriz de documentos, matriz de palavras e matriz de modificadores/qualificadores.

Na matriz de documentos, uma dimensão representa as palavras, e a outra, os documentos em que elas aparecem. Cada posição a_{ij} da matriz contém o número de vezes que a palava i aparece no documento j. Aqui, duas palavras são consideradas similares se ocorrem no mesmo documento.

Na matriz de palavras, ambas as dimensões representam as palavras. Cada posição b_{ij} da matriz contém o número de vezes que a palavra j co-ocorre com a palavra i. A co-ocorrência pode ser definida em relação a documentos, parágrafos, ou outras unidades. Aqui, duas palavras são similares se co-ocorrem com uma mesma terceira palavra.

Na matriz de modificadores, uma dimensão representa as palavras e a outra, os modificadores que aparecem modificando as palavras. Cada posição c_{ij} da matriz contém o número de vezes que a palavra j é modificada pelo modificador i. Aqui, duas palavras são consideradas similares quando são modificadas pelos mesmos modificadores.

As diferentes matrizes obtêm diferentes tipos de similaridade semântica. O tipo de informação das matrizes de documentos e de palavras captura a similaridade relacionada ao assunto, ou seja, que as palavras pertencem ao mesmo domínio. Já a informação contida na matriz de modificadores é mais detalhada, pois diferencia entidades que não têm as mesmas propriedades - diferentes propriedades correspondem a diferentes modificadores.

Estas matrizes de contagens podem ser facilmente transformadas em matrizes de probabilidades condicionais, dividindo-se cada elemento em uma linha pela soma de todos os elementos na linha. A partir disto, a similaridade semântica pode ser vista como a similaridade, ou dissimilaridade, entre duas distribuições de probabilidade.

Dagan et al. (apud [MAN99]) investigaram três medidas de dissimilaridade entre distribuições de probabilidade: a divergência de Kullback-Leibler, definida por $D(p \parallel q) = \sum_i p_i \log \frac{p_i}{q_i}$; o raio de informação, definido por $IRad(p,q) = D(p \parallel \frac{p+q}{2}) + D(q \parallel \frac{p+q}{2})$; e a norma L_1, definida por $\sum_i |p_i - q_i|$.

[17] De acordo com [JUR00], sinonímia é o nome dado ao fenômeno em que diferentes lexemas têm o mesmo significado.

A divergência de Kullback-Leibler (já vista na seção 2.2.5) mede o quão bem a distribuição q aproxima a distribuição p, ou seja, quanta informação é perdida se assumir-se a distribuição q quando a distribuição real é p. Para aplicações práticas, esta medida apresenta dois problemas:

- obtem-se o valor ∞ se há uma dimensão em que $q_i = 0$ e $p_i \neq 0$;
- a divergência de KL é assimétrica, isto é, $D(p \parallel q) \neq D(q \parallel p)$; porém, a noção intuitiva de similaridade semântica é simétrica.

A segunda medida, o raio de informação, contorna estes problemas, pois é uma medida simétrica - $IRad(p,q) = IRad(q,p)$ - e não há problema com valores infinitos, já que $\frac{p_i+q_i}{2} \neq 0$ se $p_i \neq 0$ e $q_i \neq 0$. O raio de informação pode ser interpretado como a quantidade de informação que é perdida se duas palavras, que correspondem a p e q, forem descritas com suas distribuições médias. Esta medida varia de 0, para distribuições idênticas, até $2 \log 2$, para distribuições com a máxima diferença.

A terceira medida, a norma L_1, também é simétrica e bem definida para qualquer p e q. A norma L_1 mede a proporção esperada de eventos que são diferentes nas distribuições p e q. Isto porque $\frac{1}{2}L_1(p,q) = 1 - \sum_i \min(p_i, q_i)$, e $\sum_i \min(p_i, q_i)$ é a proporção esperada de testes com o mesmo resultado.

Dagan et al. compararam as três medidas de dissimilaridade, observando a forma dos verbos como predicados dos substantivos. Dados dois verbos, as medidas de similaridade foram usadas para determinar qual o verbo adequado para um determinado substantivo. Ou seja, como a medida de similaridade é usada para computar a probabilidade condicional $P(verbo|substantivo)$:

$$P_{SIM}(v|s) = \sum_{s' \in S(s)} \frac{W(s,s')}{N(s)} P(v|s')$$

onde v é o verbo, s é o substantivo, $S(s)$ é o conjunto de substantivos mais próximos de s de acordo com a medida de similaridade, $W(s,s')$ é a medida de similaridade derivada da medida de dissimilaridade, e $N(s) = \sum_{s'} W(s,s')$ é um fator de normalização. Esta formulação necessita da transformação das medidas de dissimilaridade (KL, IRad ou L_1) em uma medida de similaridade W. As seguintes transformações são utilizadas:

$$W_{KL}(p,q) = 10^{-\beta D(p\|q)}$$
$$W_{IRad}(p,q) = 10^{-\beta IRad(p\|q)}$$
$$W_{L_1}(p,q) = (2 - L_1(p,q))^{\beta}$$

Dagan et al. mostraram que a medida IRad foi melhor que as medidas KL e L_1. Conseqüentemente, os autores recomendam IRad como a medida mais adequada para uso geral.

5 Sintaxe

Neste capítulo, são abordados os principais aspectos do processamento estatístico da linguagem natural relacionados à gramática, o que corresponde aos aspectos relacionados à estrutura sintática dos textos. Primeiramente, será apresentada a teoria dos modelos de Markov, que são utilizados em modelos estatísticos para marcação das categorias das palavras. Em seguida, serão apresentados os modelos de marcação de categorias de palavras, as gramáticas livres de contexto probabilísticas e, finalmente, a análise probabilística de corpora.

5.1 Modelos de Markov

Processos/cadeias/modelos de Markov foram inicialmente desenvolvidos por Andrei A. Markov. Sua primeira utilização foi na modelagem de seqüências de letras em trabalhos da literatura russa.

Modelos de Markov escondidos[18] são as principais ferramentas para o processamento estatístico da linguagem natural. Um HMM é uma função probabilística de um processo de Markov.

Trabalhar com a ordem das palavras nas sentenças é o que fazem os modelos visíveis de Markov[19], que serão adiante diferenciados dos HMMs. HMMs operam em um nível mais alto de abstração, inserindo estruturas "ocultas" adicionais, que permitem visualizar a ordem das categorias das palavras.

Em geral, considera-se uma seqüência, possivelmente no tempo, de variáveis aleatórias, cujos valores são dependentes dos elementos anteriores nesta seqüência. Para a maioria dos sistemas, parece razoável assumir-se que o necessário para determinar variáveis aleatória futuras é o valor da variável aleatória atual, sem a necessidade dos valores das variáveis aleatórias passadas na seqüência. Ou seja, elementos futuros da seqüência são condicionalmente independentes dos elementos passados, dados os elementos presentes.

Seja $X = (X_1, ..., X_T)$ uma seqüência de variáveis aleatórias que podem assumir os valores de algum conjunto finito $S = \{s_1, ..., s_N\}$, o espaço de estados. Desta forma, as propriedades de Markov são:

- Horizonte limitado: o valor da variável futura só depende do valor da atual.

$$P(X_{t+1} = s_k | X_1, ..., X_t) = P(X_{t+1} = s_k | X_t)$$

- Invariante no tempo (estacionário): a dependência não muda no decorrer do tempo.

$$P(X_{t+1} = s_k | X_1, ..., X_t) = P(X_2 = s_k | X_1)$$

Se X possui estas propriedades, é então considerada uma cadeia de Markov. Pode-se descrever uma cadeia de Markov através de uma matriz de transição estocástica A, onde

$$a_{ij} = P(X_{t+1} = s_j | X_{t=s_i})$$

No caso, $a_{ij} \geq 0$, $\forall i, j$ e $\sum_{j=1}^{N} a_{ij} = 1$, $\forall i$. Além disso, pode ser necessária a especificação do conjunto Π das probabilidades dos estados iniciais da cadeia de Markov:

$$\pi_i = P(X_1 = s_i)$$

Aqui, $\sum_{i=1}^{N} \pi_i = 1$. A necessidade destes valores pode ser evitada se for especificado que o modelo de Markov sempre inicia em um estado inicial extra, s_o, e utiliza transições partindo deste estado contidas na matriz A para especificar as probabilidades que ficariam registradas em Π.

Enfim, modelos de Markov podem ser usados toda vez que se quer modelar a probabilidade de uma seqüência de eventos. Alternativamente, pode-se representar uma cadeia de Markov por um diagrama de estados como na figura 2.

Os estados são representados por círculos em torno do nome do estado, e o estado inicial é indicado por uma seta de chegada. As transições possíveis são representadas por setas conectando os estados, que são rotuladas com a probabilidade de esta transição ser percorrida, dado que se está no estado de partida da seta. Transições com probabilidade zero são omitidas no diagrama. A soma das probabilidades das setas que partem de um estado é 1. A partir desta representação, pode-se dizer que um modelo de Markov é um autômato finito com probabilidades associadas aos arcos.

Em VMMs, sabe-se por quais estados da máquina se está passando. Assim, a seqüência de estados, ou alguma função determinística desta seqüência, pode ser observada.

[18] Do inglês *Hidden Markov Models* - HMMs.
[19] Do inglês *Visible Markov Models* - VMMs.

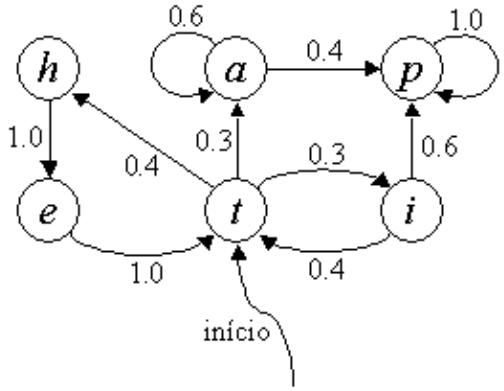

Figura 2: Um modelo de Markov.

A probabilidade de uma seqüência de estados, ou seja, de variáveis aleatórias, $X_1, ..., X_T$ é facilmente calculada para uma cadeia de Markov: multiplicam-se as probabilidades que ocorrem nos arcos ou na matriz estocástica.

$$P(X_1, ..., X_T) = P(X_1)P(X_2|X_1)P(X_3|X_2)\cdots P(X_T|X_{T-1})$$

Tratando-se do modelo de *n*-Gramas, pode-se pensar que este não é um modelo de Markov quando $n \geq 3$, pois este violaria a propriedade do horizonte limitado, já que necessita que se conheça mais de uma palavra anterior. No entanto, qualquer modelo de *n*-gramas pode ser reformulado como um VMM. Para isso, codifica-se a quantidade necessária de informação prévia no espaço de estados, sendo que cada estado é um $(n-1)$-grama. Em geral, qualquer quantidade finita de informação prévia pode ser codificada desta maneira através da elaboração do espaço de estados como um produto cartesiano[20] dos estados anteriores. Nestes casos, pode-se denominar de modelo de Markov de ordem m, onde m é o número de estados anteriores que estão sendo utilizados para prever o próximo estado. Assim, um modelo *n*-grama é equivalente a um modelo de Markov de ordem $(n-1)$.

5.1.1 Modelos de Markov escondidos

Em um HMM, não se conhece a seqüência de estados que o modelo percorre, mas apenas a seqüência de símbolos emitidos. O estado seguinte não é determinado unicamente pelos símbolos de entrada. Assim, necessita-se das probabilidades de emissão dos símbolos para cada observação:

$$P(O_t = k|X_t = s_i, X_{t+1} = s_j) = b_{ijk}$$

onde O é a seqüência de saída.

Uma razão pela qual os HMMs são muito usados é que estes podem ser eficientemente treinados usando o algoritmo para maximização da expectativa[21], que permite obter-se os parâmetros do modelo que melhor estima os dados observados.

O uso mais comum de HMMs é a tarefa de marcação de corpus, que envolve associar categorias gramaticais ou semânticas às palavras de um texto. Outro exemplo onde HMMs são bastante úteis é na geração de parâmetros para a interpolação linear de modelos *n*-gramas.

Um HMM é especificado por uma tupla (S, K, Π, A, B), onde:

- $S = \{s_1, ..., s_N\}$ é o conjunto de estados;

[20] Do inglês *crossproduct*.

[21] Algoritmo *Expectation Maximization* - EM.

- $K = \{k_1, ..., k_M\} = \{1, ..., M\}$ é o alfabeto de saída;
- $\Pi = \{\pi_i\}, i \in S$ é o conjunto das probabilidades do estado inicial;
- $A = \{a_{ij}\}, i, j \in S$ é o conjunto das probabilidades das transições de estados;
- $B = \{b_{ijk}\}, i, j \in S, k \in K$ é o conjunto das probabilidades dos símbolos de saída.

Dada a especificação de um HMM, pode-se simular o percurso de um processo de Markov e produzir uma seqüência de saída $O = (o_1, ..., o_T), o_t \in K$. Para isso, pode-se utilizar o algoritmo mostrado na figura 3.

```
t:= 1;
Inicie no estado s_i com probabilidade π_i
para sempre faça
  Mova-se do estado s_i para o estado s_j com probabilidade a_ij
  Emita o símbolo o_t = k com probabilidade b_ijk
  t:= t+1
end
```

Figura 3: Um algoritmo para simular um proceso de Markov.

No entanto, mais interessante que a simulação é assumir que algum conjunto de dados foi gerado pelo HMM e, então, ser capaz de calcular probabilidades e a provável seqüência de estados percorrida.

5.1.2 Questões fundamentais referentes a HMMs

Há três questões fundamentais sobre um HMM:

1. Dado um modelo $\mu = (A, B, \Pi)$, como computar eficientemente a probabilidade de uma observação, isto é, $P(O|\mu)$?
2. Dada a seqüência de observações O e um modelo μ, como escolher uma seqüência de estados $(X_1, ..., X_{T+1})$ que melhor descreve as observações?
3. Dada uma seqüência de observações O e um espaço de possíveis modelos, encontrados variando-se os parâmetros do modelo $\mu = (A, B, \Pi)$, como encontrar o modelo que melhor descreve os dados observados?

A primeira questão pode ser usada para decidir qual o melhor em um conjunto de modelos. A segunda questão leva a identificar qual o que caminho seguido através da cadeia de Markov, e este caminho oculto pode ser usado para classificação. A terceira questão mostra que não se conhece os parâmetros e tem-se que estimá-los a partir dos dados.

Nas subseções a seguir, serão mostradas técnicas para encontrar a probabilidade de uma observação, encontar a melhor seqüência de estados que descreve uma observação, e estimar os parâmetros de um modelo que melhor descrevem os dados observados.

5.1.2.1 Encontrando a probabilidade de uma observação

Dada uma seqüência de observações $O = (o_1, ..., o_T)$ e um modelo $\mu = (A, B, \Pi)$, deseja-se saber como computar eficientemente $P(O|\mu)$, que corresponde à probabilidade da observação.

Para qualquer seqüência de estados $X = (X_1, ..., X_{T+1})$, tem-se:

$$P(O|X, \mu) = \prod_{t=1}^{T} P(o_t|X_t, X_{t+1}, \mu) = b_{X_1 X_2 o_1} b_{X_2 X_3 o_2} \cdots b_{X_T X_2 o_1}$$

e

$$P(X|\mu) = \pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \cdots a_{X_T X_{T+1}}$$

Como

$$P(O, X|\mu) = P(O|X, \mu) P(X|\mu)$$

então tem-se

$$P(O|\mu) = \sum_X P(O|X, \mu) P(X|\mu) = \sum_{X_1 \cdots X_{T+1}} \pi_{X_1} \prod_{t=1}^{T} a_{X_t X_{t+1}} b_{X_t X_{t+1} o_t}$$

Simplesmente, somam-se as probabilidades de cada observação ocorrida, de acordo com cada possível seqüência de estados. No entanto, se esta expressão for avaliada em relação ao caso geral, é extremamente ineficiente, pois requer $(2T+1)N^{T+1}$ multiplicações. Para evitar esta complexidade, pode-se utilizar a técnica de programação dinâmica, que mantem resultados parciais, ao invés de recalculá-los. Para algoritmos como os HMMs, o problema de programação dinâmica é geralmente descrito em termos de "grades" - faz-se uma matriz quadrada de estados versus tempo, e computam-se as probabilidades de estar em cada estado, no instante de tempo anterior. Uma grade pode armazenar a probabilidade de todos os subcaminhos iniciais do HMM, que terminam em um certo estado, em um certo tempo. A probabilidade de subcaminhos mais longos pode ser calculada em termos de subcaminhos menores.

Pode-se descrever este processo através de variáveis *forward*, utilizando o algoritmo *forward*, descrito a seguir.

O procedimento *forward*. Na posição (s_i, t) da grade, é armazenada uma variável *forward* $\alpha_i(t) = P(o_1 o_2 \cdots o_{t-1}, X_t = i|\mu)$, que expressa a probabilidade total de estar-se no estado s_i no tempo t. A mesma é calculada somando-se as probabilidades de todos os arcos que chegam a um nodo da grade. Calcula-se as variáveis *forward* da grade da esquerda para a direita, usando o seguinte algoritmo:

1. Inicialização: $\alpha_i(1) = \pi_i, 1 \leq i \leq N$;
2. Indução: $\alpha_j(t+1) = \sum_{i=1}^{N} \alpha_i(t) a_{ij} b_{ij o_t}$;
3. Total: $P(O|\mu) = \sum_{i=1}^{N} \alpha_i(T+1)$.

Este é um algoritmo de baixo custo, que requer apenas $2N^2T$ multiplicações.

O procedimento *backward*. O procedimento *backward* computa as variáveis *backward* $\beta_i(t) = P(o_t \cdots o_T|X_t = i, \mu)$, que correspondem à probabilidade total de conhecer-se o restante da seqüência de observações, sendo que se está no estado s_i no tempo t. As probabilidades *backward*, combinadas com as probabilidades *forward*, são importantes para a solução do problema de reestimação dos parâmetros.

Variáveis *backward* podem ser calculadas pelo seguinte algoritmo, da direita para a esquerda na grade:

1. Inicialização: $\beta_i(T+1) = 1, 1 \leq i \leq N$;
2. Indução: $\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_{ij o_t} \beta_j(t+1), 1 \leq t \leq T, 1 \leq i \leq N$;
3. Total: $P(O|\mu) = \sum_{i=1}^{N} \pi_i \beta_i(1)$.

Combinando os procedimentos *forward* e *backward*. Na maioria dos casos, de acordo com [MAN99], utiliza-se uma combinação dos procedimentos *forward* e *backward* para obter-se as probabilidades de uma seqüência de observações. Assim, tem-se:

$$P(O|\mu) = \sum_{i=1}^{N} \alpha_i(t) \beta_i(t), 1 \leq t \leq T+1$$

5.1.2.2 Encontrando a melhor seqüência de estados

O algoritmo mais utilizado para encontrar a seqüência de estados que melhor descreve uma observação é o algoritmo de Viterbi, que computa a seqüência de estados mais provável.

O algoritmo de Viterbi. Deseja-se encontrar o caminho completo mais provável que é $\arg\max_X P(X|O,\mu)$. Para isto, é suficiente maximizar $\arg\max_X P(X|O,\mu)$ para um O fixo. O algoritmo de Viterbi, que é um algoritmo de grade que define:

$$\delta_j(t) = \max_{X_1 \cdots X_{t-1}} P(X_1 \cdots X_{t-1}, o_1 \cdots o_{t-1}, X_t = j|\mu)$$

Para cada ponto na grade, $\delta_j(t)$ registra a probabilidade do caminho mais provável que leva ao nodo. A variável correspondente $\psi_j(t)$ armazena o nodo do arco de chegada que levou a este caminho mais provável, para poder-se posteriormente traçar o caminho percorrido. Usando programação dinâmica, calcula-se o caminho mais provável, utilizando a grade, da seguinte forma:

1. Inicialização: $\delta_j(1) = \pi_j, 1 \leq j \leq N$;
2. Indução: $\delta_j(t+1) = \max_{1\leq j\leq N} \delta_i(t)a_{ij}b_{ijo_t}, 1 \leq j \leq N$, e $\psi_j(t+1) = \arg\max_{1\leq i\leq N} \delta_i(t)a_{ij}b_{ijo_t}, 1 \leq j \leq N$;
3. Término e leitura do caminho (por *backtracking*). A seqüência de estados mais provável é obtida de traz pra frente: $\widehat{X}_{T+1} = \arg\max_{1\leq i\leq N} \delta_i(T+1)$; $\widehat{X}_t = \psi_{\widehat{X}_{t+1}}(t+1)$, e $P(\widehat{X}) = \max_{1\leq i\leq N} \delta_i(T+1)$.

Em aplicações práticas, comumente trabalha-se com as n melhores seqüências, ou seja, um grafo dos caminhos mais prováveis.

5.1.2.3 Estimativa de parâmetros - treinando um HMM

Dada uma certa seqüência de observações, quer-se encontrar os valores dos parâmetros do modelo $\mu = (A, B, \Pi)$ que melhor descrevem os dados observados. Utilizando a estimação da probabilidade máxima, quer-se encontrar os valores que maximizam $P(O|\mu)$: $\arg\max_\mu P(O_{training}|\mu)$.

Não há nenhum método analítico conhecido para selecionar o modelo μ que maximize $P(O|\mu)$. Porém, esta probabilidade pode ser maximizada localmente por um algoritmo iterativo conhecido como algoritmo *forward-backward* ou Baum-Welch, que é um caso especial do método de maximização da expectativa.

Algoritmo *Forward-Backward*. Este algoritmo, dada uma seqüência de observações, ajusta as probabilidades dos parâmetros de um HMM, afim de que a seqüência em questão torne-se a mais provável possível. Este algoritmo utiliza as variáveis *forward* e *backward*, apresentadas anteriomente na seção 5.1.2.1.

Não se conhece o modelo que gerou determinada seqüência de saída, mas pode-se obter a probabilidade desta seqüência ocorrer usando-se algum modelo escolhido aleatoriamente. Calculada a seqüência de observações de um certo modelo, pode-se observar quais transições de estados e símbolos emitidos foram provavelmente mais usados. Assim, incrementando-se a probabilidade destes itens, pode-se optar por um modelo revisado, que resulte em uma maior probabilidade para a seqüência de observações.

Este processo de maximização de probabilidades é freqüentemente denominado treinamento do modelo, e executado sobre dados de treinamento.

A probabilidade de passar por um certo arco no tempo t, dada a seqüência de observações O, é definida por $p_t(i,j)$, $1 \leq t \leq T$, $1 \leq i,j \leq N$:

$$p_t(i,j) = P(X_t = 1, X_{t+1} = j|O,\mu) = \frac{P(X_t = i, X_{t+1} = j, O|\mu)}{P(O|\mu)} = \frac{\alpha_i(t)a_{ij}b_{ijo_t}\beta_j(t+1)}{\sum_{m=1}^{N} \alpha_m(t)\beta_m(t)}$$

Seja $\gamma_i(t) = \sum_{j=1}^{N} p_t(i,j)$. Se acumular-se no tempo, obtém-se as seguintes expectativas:

- $\sum_{t=1}^{T} \gamma_i(t)$ = número esperado de transições do estado i em O;
- $\sum_{t=1}^{T} p_t(i,j)$ = número esperado de transições do estado i para o estado j em O.

Inicia-se com um modelo μ (pré-selecionado ou escolhido aleatóriamente). Computa-se O através do modelo escolhido para estimar as expectativas de cada parâmetro do modelo. Então, altera-se o modelo para maximizar os valores dos caminhos mais usados. O processo se repete até convergir para valores ótimos para os parâmetros de μ. As fórmulas para reestimação são as seguintes:

- $\widehat{\pi}_i$ = (freqüência esperada no estado i no tempo $t = 1$) = $\gamma_i(1)$;
- $\widehat{a}_{ij}$ = (número esperado de transições do estato i para o j / número esperado de transições do estado i) = $\frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{t=1}^{T} \gamma_i(t)}$;
- $\widehat{b}_{ij}$ = (número esperado de transições do estado i para o j com k observado) = $\frac{\sum_{\{t:o_t=k, 1 \leq t \leq T\}} p_t(i,j)}{\sum_{t=1}^{T} p_t(i,j)}$.

Assim, a partir de $\mu = (A, B, \Pi)$, obtém-se $\widehat{\mu} = (\widehat{A}, \widehat{B}, \widehat{\Pi})$. Baum provou que $P(O|\widehat{\mu}) \geq P(O|\mu)$.

Normalmente, a reestimação de parâmetros continua até que os resultados estabilizem. O processo de reestimação não garante que será encontrado o melhor modelo, pois pode-se parar em um máximo local. No entanto, a reestimação de Baum-Welch é geralmente eficiente para HMMs.

5.2 Marcação de corpora

A marcação de uma sentença pode ser definida como a tarefa de rotular cada palavra da sentença com sua categoria morfo-sintática [MAN99]. Dada uma seqüência de palavras e um conjunto de rótulos, associa-se a cada palavra o rótulo correspondente. Como muitas palavras têm mais de uma categoria sintática, a marcação visa determinar qual destas categorias é a mais provável para um uso particular da palavra na sentença.

Desta forma, o objetivo de um sistema marcador/rotulador, conforme [VIL95], é encontrar a seqüência mais provável de marcas/rótulos que correspondam à seqüência de palavras dada. Por isso, a marcação pode ser vista como uma tarefa que reduz a ambigüidade das sentenças.

O processo de marcação do corpus é um nível intermediário de representação, que é útil e mais tratável que a análise sintática completa.

Há essencialmente duas principais fontes de informação para decidir a categoria de uma palavra em um contexto:

- as marcas das outras palavras do contexto, que permitem a observação de seqüências comuns de categorias de palavras. Este tipo de informação estrutural sintagmática é a mais visível através da marcação, mas isolada não é de grande importância;
- as informações léxicas sobre a própria palavra, pois palavras com mais de uma categoria ocorrem, na maioria das vezes, com uma das suas categorias possíveis. Esta categoria mais freqüente é dita básica, e as demais são vistas como derivadas da categoria básica.

A eficiência de um sistema para marcação de corpora depende consideravelmente dos seguintes fatores:

- a quantidade de dados de treinamento disponível: em geral, quanto mais dados, melhor;
- o conjunto de rótulos: normalmente, quanto maior o conjunto de rótulos, maior a possibilidade de ambigüidade, dificultando a tarefa de marcação;

- ◇ a diferença entre dados de treinamento e dados da aplicação: se o corpus de treinhamento e de aplicação são obtidos da mesma fonte, a precisão da marcação será maior. Se o corpus da aplicação for de uma fonte diferente ou de um gênero diferente de texto, a performance do sistema de marcação será pior;
- ◇ palavras desconhecidas: a ocorrência de muitas palavras desconhecidas no corpus pode decradar consideravelmente a performance da marcação.

De acordo com [MAN99], estes tipos de condições externas freqüentemente têm maior influência na performance do que a escolha do método de marcação a ser implementado.

Conforme [JUR00], a maioria dos algoritmos para marcação podem ser classificados em um de dois tipos: baseados em regras ou estatísticos.

Os rotuladores baseados em regras geralmente envolvem uma grande base de regras para resolução da ambigüidade. Estas regras especificam, por exemplo, as situações em que certa palavra ambígua é um substantivo e não um verbo. Os rotuladores baseados em regras não serão aqui mais profundamente abordados.

Os rotuladores estatísticos geralmente resolvem ambigüidades de marcação através do uso de um corpus de treinamento para computar a probabilidade de uma dada palavra ter um determinado rótulo, dado o contexto em que se encontra.

Há diferentes modelos estatísticos propostos para realizar a tarefa de marcação das categorias das palavras de um corpus. A seguir serão apresentados rotuladores baseados nos modelos de Markov, que requerem um conjunto de dados de treinamento marcados manualmente, e rotuladores baseados nos modelos de Markov escondidos, que são apropriados para situações em que não se possui dados de treinamento.

Ainda há outra abordagem de rotuladores, os rotuladores baseados em transformações, que compartilham características dos dois modelos de marcação anteriores. O modelo de marcação baseado em transformações será apresentado na seção 5.2.3.

5.2.1 Rotuladores baseados no modelo de Markov

Em rotuladores baseados no modelo de Markov, a seqüência de rótulos em um texto é vista como uma cadeia de Markov, que tem as propriedades de horizonte limitado e invariância no tempo, descritas na seção 5.1.

Isto significa que se assume que o rótulo de uma palavra dependa somente do rótulo da palavra anterior (horizonte limitado) e que esta dependência não mude com o passar do tempo (invariância no tempo). Por exemplo, se um verbo tem probabilidade 0.2 de ocorrer depois de um pronome, no início da sentença, então esta probabilidade muda conforme for feita a marcação do resto da sentença ou das próximas sentenças.

Sejam w_i a palavra na posição i do corpus, t_i o rótulo da palavra w_i, $w_{i,j}$ as palavras que ocorrem da posição i até a posição j no corpus, e $t_{i,j}$ os rótulos das palavras $w_{i,j}$. Assim, pode-se reescrever a propriedade do horizonte limitado como:

$$P(t_{i+1}|t_{1,i}) = P(t_{i+1}|t_i)$$

Utiliza-se um conjunto de treinamento que contém textos marcados manualmente, para que as regularidades das seqüências de rótulos possam ser identificadas. A estimativa da probabilidade máxima de o rótulo t^k (rótulo de número k no conjunto de rótulos) seguir o rótulo t^j (rótulo de número j no conjunto de rótulos) é calculada a partir das freqüências relativas de diferentes rótulos seguindo um determinado rótulo. Assim, tem-se:

$$P(t^k|t^j) = \frac{C(t^j, t^k)}{C(t^j)}$$

onde $C(t^j, t^k)$ é o número de ocorrências de t^j seguindo t^k, e $C(t^j)$ é o número de ocorrências de t^j nos dados de treinamento.

Com as estimativas das probabilidades $P(t_{i+1}|t_i)$, pode-se computar a probabilidade de uma seqüência de rótulos. Na prática, a tarefa consiste em encontrar a seqüência de rótulos mais provável para uma dada

seqüência de palavras, isto é, encontrar a seqüência de estados para a seqüência de palavras, sendo que os estados do modelo de Markov correspondem aos rótulos.

Como pode-se observar os estados/rótulos diretamente, já que tem-se um corpus marcado, é possível estimar a probabilidade de uma palavra w^l (a palavra de número l no dicionário) ser emitida por um estado particular, através da estimação da probabilidade máxima:

$$P(w^l|t^j) = \frac{C(w^l, t^j)}{C(t^j)}$$

Então, para encontrar-se a melhor marcação $t_{1,n}$ para uma sentença $w_{1,n}$, primeiramente aplica-se a regra de Bayes e tem-se:

$$\arg\max_{t_{1,n}} P(t_{1,n}|w_{1,n}) = \arg\max_{t_{1,n}} \frac{P(w_{1,n}|t_{1,n})P(t_{1,n})}{P(w_{1,n})} = \arg\max_{t_{1,n}} P(w_{1,n}|t_{1,n})P(t_{1,n})$$

Esta expressão pode ser reduzida a parâmetros que podem ser estimados a partir do corpus de treinamento. Além de supor-se a propriedade do horizonte limitado, pode-se fazer mais duas suposições sobre as palavras:

- palavras são independentes umas das outras:

$$P(w_{1,n}|t_{1,n})P(t_{1,n}) = \prod_{i=1}^{n} P(w_i|t_{1,n}) \times P(t_n|t_{1,n-1}) \times P(t_{n-1}|t_{1,n-2}) \times ... \times P(t_2|t_1)$$

- a identidade de uma palavra só depende de seu rótulo:

$$P(w_{1,n}|t_{1,n})P(t_{1,n}) = \prod_{i=1}^{n} P(w_i|t_i) \times P(t_n|t_{n-1}) \times P(t_{n-1}|t_{n-2}) \times ... \times P(t_2|t_1)$$

A equação final para determinar a melhor marcação para uma sentença é:

$$\widehat{t}_{1,n} = \arg\max_{t_{1,n}} P(t_{1,n}|w_{1,n}) = \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

A figura 4 mostra uma simplificação do algoritmo de treinamento de um rotulador baseado no modelo de Markov.

```
para todos os rótulos t^j faça
  para todos os rótulos t^k faça
    P(t^k|t^j) := C(t^j,t^k) / C(t^j)
  end
end
para todos os rótulos t^j faça
  para todas as palavras w^l faça
    P(w^l|t^j) := C(w^l,t^k) / C(t^j)
  end
end
```

Figura 4: Algoritmo para treinamento de um rotulador baseado no modelo de Markov visível.

Tem-se, então, um modelo de Markov visível treinado. No entanto, este será tratado como um modelo de Markov escondido na tarefa de marcação de um corpus, que pode ser realizada pelo algoritmo de Viterbi (visto em 5.1.2.2).

5.2.1.1 Algoritmo de Viterbi para marcação

Quer-se obter a melhor seqüência de rótulos $t_{1,n}$ para uma sentença de tamanho n. Sabe-se que os estados representam os rótulos.

Como visto anteriormente, o algoritmo de Viterbi computa duas funções: $\delta_i(j)$, que resulta a probabilidade de estar no estado j (rótulo j) com a palavra i; e $\psi_{i+1}(j)$, que resulta o estado (rótulo) mais provável para a palavra i, dado que se está no estado j, com a palavra $i+1$.

No passo de inicialização do algoritmo, atribui-se probabilidade 1.0 ao primeiro rótulo da sentença, e probabilidade 0.0 aos demais rótulos para a primeira posição da sentença.

No passo de indução, onde $a_{jk} = P(t^k|t^j)$ e $b_{jkw^l} = P(w^l|t^j)$, calcula-se $\delta_{i+1}(t^j)$ e $\psi_{i+1}(t^j)$, com base na equação para $\widehat{t}_{1,n}$:

$$\delta_{i+1}(t^j) = \max_{1 \leq j \leq T} [\delta_i(t^k) \times P(w_{i+1}|t^j) \times P(t^j|t^k)], \quad 1 \leq j \leq T$$

$$\psi_{i+1}(t^j) = \arg \max_{1 \leq j \leq T} [\delta_i(t^k) \times P(w_{i+1}|t^j) \times P(t^j|t^k)], \quad 1 \leq j \leq T$$

No passo de término e leitura do caminho, onde $X_1, ..., X_n$ são os rótulos escolhidos para as palavras $w_1, ..., w_n$, calcula-se:

$$X_n = \arg \max_{1 \leq j \leq T} \delta_n(t^j)$$

$$X_i = \psi_{i+1}(X_{i+1}), \quad 1 \leq i \leq n-1$$

$$P(X_1, ..., X_n) = \max_{1 \leq j \leq T} \delta_{n+1}(t^j)$$

5.2.1.2 Rotuladores de trigramas

Um rotulador básico baseado no modelo de Markov pode ser estendido de várias maneiras. No modelo apresentado até o momento, são feitas predições baseadas apenas no rótulo precedente. Este é dito um rotulador de bigramas, pois as unidades básicas consideradas são o rótulo anterior e o rótulo corrente.

No entanto, pode-se obter predições mais confiáveis se um contexto maior for considerado. Um rotulador de trigramas considera os dois rótulos prececentes, possibilitando a resolução da ambigüidade de um maior número de casos.

5.2.2 Rotuladores baseados no modelo de Markov escondido

Os rotuladores baseados no modelo de Markov visível necessitam de um conjunto de dados de treinamento rotulados, que deve ser grande para que o rotulador seja eficiente. No entanto, grandes quantidades de dados rotulados não são facilmente encontradas.

Já os rotuladores baseados no modelo de Markov escondido não necessitam de dados de treinamento. Utiliza-se um HMM para aprender as regularidades das seqüências de rótulos.

Como no caso do VMM, os estados do HMM correspondem aos rótulos. O alfabeto de saída consiste das palavras do dicionário ou de classes de palavras (pode-se agrupar as palavras em classes de equivalência, assim todas as palavras que admitem o mesmo conjunto de rótulos são agrupadas na mesma classe).

Os parâmetros do HMM são inicializados com base em informações obtidas no dicionário. À probabilidade de emissão de um símbolo (no caso, corresponde a probabilidade de geração de uma palavra, já que o alfabeto de saída é composto por palavras) é atribuído o valor 0, se o par palavra-rótulo correspondente não está listado no dicionário.

Denota-se por $b_{j,l}$ a probabilidade de a palavra (ou classe de palavras) l ser emitida pelo estado que representa o rótulo j. Isto significa que, como no caso do VMM, a saída de um estado/rótulo não depende do estado/rótulo seguinte.

Dentre os métodos existentes para o cálculo da probabilidade de geração de uma palavra, a partir de um rótulo, serão apresentados os métodos propostos por Jelinek e Kupiec.

O método de Jelinek inicializa o HMM com a estimativa da probabilidade máxima para $P(w^k, t^i)$, assumindo que a ocorrência das palavras é igualmente provável com cada um de seus rótulos possíveis. Calcula-se $b_{j.l}$ da seguinte forma:

$$b_{j.l} = \frac{b^*_{j.l} C(w^l)}{\sum_{w^m} b^*_{j.m} C(w^m)}$$

onde o somatório é feito sobre todas as palavras w^m do dicionário, e

$$b^*_{j.l} = \begin{cases} 0 & \text{se } t^j \text{ não é um rótulo admitido por } w^l \\ \frac{1}{T(w^l)} & \text{caso contrário} \end{cases}$$

onde $T(w^l$ é o número de rótulos admitidos por w^l.

O método de Kupiec, primeiramente, agrupa todas as palavras que admitem os mesmos rótulos em "meta-palavras" u_L. L é o subconjunto dos inteiros de 1 a T, onde T é o número de rótulos diferentes no conjunto de rótulos.

$$u_L = \{w^l | j \in L \leftrightarrow t^j \text{ é admitido por } w^l\} \quad \forall L \subseteq \{1, ..., T\}$$

As meta-palavras u_L são tratadas da mesma maneira que as palavras no método de Jelinek.

$$b_{j.L} = \frac{b^*_{j.L} C(u_L)}{\sum_{u_{L'}} b^*_{j.L'} C(u_{L'})}$$

onde $C(u_L)$ é o número de ocorrências das palavras de u_L, o somatório é feito sobre as meta-palavras $u_{L'}$, e

$$b^*_{j.L} = \begin{cases} 0 & \text{se } j \notin L \\ \frac{1}{|L|} & \text{caso contrário} \end{cases}$$

onde $|L|$ é o número de índices de L.

A vantagem do método de Kupiec é que este não precisa ajustar um conjunto de parâmetros para cada palavra, pois, com o uso das classes de equivalência, o número total de parâmetros é reduzido e pode-se estimá-los com mais confiança. No entanto, este aspecto do método de Kupiec pode tornar-se uma desvantagem quando há dados de treinamento suficientes para estimar, com precisão, os parâmetros palavra por palavra, como faz o método de Jelinek.

Feita a inicialização do HMM, este é treinado através do algoritmo *Forward-Backward*, apresentado na seção 5.1.2.3.

A marcação do corpus é realizada pelo algoritmo de Viterbi, da mesma forma que nos rotuladores baseados no modelo de Markov visível (subseção 5.2.1.1).

Em [VIL95], é proposto um rotulador estatístico de categorias morfo-sintáticas para a língua portuguesa. Este sistema é composto de três módulos principais: o módulo classificador, o módulo construtor de HMMs, e o módulo de Viterbi.

O módulo classificador recebe um corpus de entrada, decompõe cada sentença do corpus em palavras, e atribui a cada palavra uma classe de equivalência, conforme definido no dicionário utilizado, como no método de Kupiec. Com o resultado do módulo classificador, rotula-se o corpus manualmente, com base nas classes de equivalência das palavras.

O módulo construtor de HMMs é responsável por construir o HMM a partir da análise dos padrões lingüísticos que ocorrem no corpus de treinamento marcado com rótulos e classes de equivalência. Então, são estimados os parâmetros do HMM .

O módulo de Viterbi aplica o algoritmo de Viterbi sobre o HMM treinado.

Em [THE99], é apresentada uma extensão do modelo de Markov escondido para marcação de categorias das palavras, utilizando aproximações de segunda ordem para probabilidades contextuais e lexicais. Estas aproximações fazem com que sejam utilizadas mais informações contextuais do que o padrão em sistemas estatísticos. Este novo tipo de rotulador utiliza trigramas, não somente para probabilidades contextuais, mas também para propriedades léxicas. Refere-se a este modelo como um HMM de segunda-ordem completa.

5.2.3 Rotuladores baseados em transformações

A marcação de corpora baseada em tranformações é uma instância do aprendizado baseado em transformações[22] e compartilha características dos rotuladores baseados em regras e dos rotuladores estatísticos. Como os rotuladores baseados em regras, o TBL é baseado em regras que especificam quais rótulos devem ser associados a quais palavras. E como os rotuladores estatísticos, TBL é uma técnica de aprendizado automático em que as regras são induzidas automaticamente a partir dos dados. TBL é uma técnica supervisionada de aprendizado, isto é, pressupõe um corpus de treinamento já marcado.

O algoritmo TBL possui um conjunto de regras de marcação. Antes da aplicação das regras, cada palavra é marcada com o rótulo mais provável, obtido a partir de um corpus marcado ou de um dicionário.

Então, as regras de transformação podem ser aplicadas. É feito um *ranking* entre as regras para determinar a sua ordem de aplicação. Primeiramente, o corpus é marcado pela utilização da regra mais bem colocada no *ranking*. Em seguida, a regra seguinte, um pouco mais específica, altera algumas das marcações anteriores. Então, repete-se o processo até utilizar-se a regra mais específica, que altera o menor número de rótulos.

5.2.3.1 O algoritmo de aprendizado

O algoritmo de aprendizado da marcação baseada em transformações seleciona as melhores regras de transformação e determina a sua ordem de aplicação.

Em cada iteração, o algoritmo escolhe a transformação que reduz a taxa de erro, que é medida pelo número de palavras marcadas incorretamente. O critério de parada é não restar mais nenhuma regra que minimize a taxa de erro.

5.3 Gramáticas livres de contexto probabilísticas

Modelos de *n*-gramas e HMMs permitem apenas que se processe as sentenças linearmente. No entanto, até as sentenças mais simples requerem um modelo não linear que reflita sua estrutura hierárquica ao invés da ordem linear das palavras.

Nas formas mais tradicionais de gramática, a sintaxe é representada não só pela ordem linear das palavras, mas também pela forma como as palavras se agrupam e se relacionam. Como as linguagens têm uma estrutura recursiva complexa, tem-se utilizado modelos baseados em árvores para representar as sentenças.

Gramáticas livres de contexto probabilísticas[23] são, conforme [MAN99], o mais simples e natural modelo probabilístico para tratar estruturas de árvores, e seus algoritmos são semelhantes aos utilizados para HMMs. As PCFGs podem ser utilizadas na construção de analisadores sintáticos de linguagens.

Uma PCFG é uma gramática livre de contexto com probabilidades adicionadas às suas regras, indicando a probabilidade de utilização de cada regra. Uma PCFG consiste de:

- um conjunto de símbolos terminais, $\{w^k\}$, $k = 1, ..., V$;
- um conjunto de símbolos não terminais, $\{N^i\}$, $i = 1, ..., n$;
- um símbolo inicial, N_1;

[22] Do inglês *transformation-based learning* - TBL.
[23] Do inglês *Probabilistic Context Free Grammar - PCFG.*

- um conjunto de regras, $\{N^i \rightarrow \zeta^j\}$, onde ζ^j é uma seqüência de símbolos terminais e não terminais;
- um conjunto de probabilidades correspondentes às regras, tal que $\forall i \quad \sum_j P(N^i \rightarrow \zeta^j) = 1$.

A figura 5 mostra um exemplo de PCFG que é capaz de reconhecer a sentença *Professores olhavam meninos com binóculos.* Nesta figura, S (sentença), NP (sintagma nominal), VP (sintagma verbal), PP (sintagma preposicional), P (preposição) e V (verbo) são os símbolos não terminais. O símbolo inicial N^1 é representado por S. Os símbolos terminais são *professores*, *meninos*, *binóculos*, *olhavam* e *com*.

S $\rightarrow$ NP VP	1.0	V $\rightarrow$ *olhavam*	1.0
PP $\rightarrow$ P NP	1.0	NP $\rightarrow$ *professores*	0.2
VP $\rightarrow$ V NP	0.7	NP $\rightarrow$ *meninos*	0.2
VP $\rightarrow$ VP PP	0.3	NP $\rightarrow$ *binóculos*	0.1
NP $\rightarrow$ NP PP	0.5	P $\rightarrow$ *com*	1.0

Figura 5: Um exemplo de PCFG.

A soma das probabilidades de todas as regras que geram o mesmo não terminal deve ser 1. Intuitivamente, $P(N^i \rightarrow \zeta^j)$ significa a probabilidade de gerar o não terminal N^i utilizando esta regra ao invés de qualquer outra regra para N^i.

As sentenças a serem analisadas serão representadas pela seqüência de palavras $w_1 \cdots w_m$, notada por $w_{1,m}$. Uma subseqüência de palvras $w_a \cdots wb$ será notada por $w_{a,b}$. Se, a partir do resultado de uma ou mais operações de derivação na gramática, for possível reescrever o não terminal N^j como uma seqüência de palavras $w_{a,b}$, então diz-se que N^j domina as palavras $w_{a,b}$, e escreve-se $N^j \stackrel{*}{\Longrightarrow} w_{a,b}$ ou $\text{produção}(N^j) = w_{a,b}$. Para dizer-se que um não terminal engloba as posições a até b na seqüência, mas não especifica quais palavras estão contidas nesta subseqüência, escreve-se $N^j_{a,b}$.

A probabilidade de uma sentença, de acordo com uma gramática G, é dada por:

$$P(w_{1,m}) = \sum_t P(w_{1,m}, t) = \sum_{\{t:\text{produção}(t)=w_{1,m}\}} P(t)$$

onde t é uma árvore de derivação da sentença.

Para encontar-se a probabilidade de uma árvore de derivação, basta multiplicar as probabilidades das regras que formam suas subárvores. As condições para isso são:

- não variar de acordo com o lugar: a probabilidade de uma subárvore não depende do lugar em que estão, na seqüência, as palavras que esta domina (equivale à propriedade de invariância no tempo dos HMMs): $\forall k \quad P(N^j_{k,k+c} \rightarrow \zeta)$ é a mesma;
- ser livre de contexto: a probabilidade de uma subárvore não depende das palavras não dominadas por ela: $P(N^j_{k,l} \rightarrow \zeta | \text{quaisquer palavras fora de } k \text{ e } l) = P(N^j_{k,l} \rightarrow \zeta)$;
- ser livre de antecessor: a probabilidade de uma subárvore não depende dos nodos de derivação que estejam fora da subárvore: $P(N^j_{k,l} \rightarrow \zeta | \text{qualquer nodo antecessor fora de } N^j_{k,l}) = P(N^j_{k,l} \rightarrow \zeta)$.

Serão aqui consideradas somente gramáticas que estão de acordo com a forma normal de Chomsky, que têm apenas regras unárias e binárias, nas formas:

$$N^i \rightarrow N^j \; N^k$$
$$N^i \rightarrow w^j$$

Os parâmetros de uma PCFG na forma normal de Chomsky são:

$$P(N^j \rightarrow N^r \; N^s | G) \quad \text{se há } n \text{ não terminais, então haverá uma matriz de } n^3 \text{ parâmetros}$$
$$P(N^j \rightarrow w^k | G) \quad \text{se há } V \text{ terminais, então haverá } nV \text{ parâmetros}$$

Para $j = 1, ..., n$, tem-se:

$$\sum_{r,s} P(N^j \rightarrow N^r\ N^s) + \sum_k P(N^j \rightarrow w^k) = 1$$

A gramática da figura 5 satisfaz esta restrição.

Há semelhanças significativas entre as PCFGs e os HMMs. As probabilidades das regras de uma PCFG podem ser calculadas a partir de conhecimentos já vistos quando tratou-se dos HMMs.

5.3.1 De HMMs a PCFGs

De acordo com [CHA93], a melhor maneira de verificar as similaridades entre HMMs e PCFGs é considerar um tipo mais simples de gramáticas, as gramáticas regulares probabilísticas[24].

Uma PRG tem um estado inicial N^1 e as regras na seguinte forma: $N^i \rightarrow w^j\ N^k$ ou $N^i \rightarrow w^j$.

Cada não terminal de uma PRG pode ser visto como um estado de um HMM, e cada regra na forma $N^i \rightarrow w^j\ N^k$ pode ser vista como uma transição do estado N^i ao estado N^k, com emissão do terminal w^j. Note-se que não há nenhum aspecto dos HMMs que corresponda às regras de uma PRG na forma $N^i \rightarrow w^j$. Isto ocorre porque os dois conceitos, de PRG e HMM, não são exatamente iguais. A diferença está na maneira como se associam as probabilidades. Uma gramática probabilística associa uma probabilidade a cada sentença de uma linguagem, tal que a soma das probabilidades de todas as sentenças seja 1. Já os HMMs consideram seqüências de um tamanho n, tais que a soma das probabilidades de todas as seqüências de tamanho n seja 1. Ou seja, enquanto, em relação aos HMMs, tem-se uma distribuição de probabilidade sobre seqüências de um certo tamanho - $\forall n\ \sum_{w_{1,n}} P(w_{1,n}) = 1$, em relação a PRGs e PCFGs, tem-se:

$$\sum_{\omega \in \mathcal{L}} P(\omega) = 1$$

onde $\mathcal{L}$ é a linguagem gerada pela gramática.

Então, pode-se ver uma PRG como um HMM com um estado inicial e um estado final. O estado inicial se liga aos estados do HMM com as probabilidades iniciais definidas em Π. O estado final representa o fim da sentença e, uma vez alcançado, não se pode mais sair. De cada estado do HMM, pode-se continuar percorrendo os estados do HMM ou ir ao estado final, que é interpretado como o fim da sentença na PRG.

Pode-se implementar uma PRG a partir de um HMM, onde os estados são os não terminais e os símbolos de saída são os terminais. Nos HMMs, podia-se utilizar as probabilidades *forward* $\alpha_i(t)$ e *backward* $\beta_i(t)$, vistas na seção 5.1.2.1, retomadas da seguinte forma:

$$\alpha_i(t) = P(w_{1,t-1}, X_t = i)$$
$$\beta_i(t) = P(w_{t,T} | X_t = i)$$

Em uma árvore de derivação, a probabilidade *backward* corresponde à probabilidade de tudo que está abaixo de um determinado nodo, ou seja, o que é dominado pelo nodo. Como, para determinar o domínio de um nodo N^j, são necessários dois argumentos, então a probabilidade *backward* pode ser convertida na probabilidade *inside* $\beta_j(p,q)$, em uma abordagem para tratar com as PCFGs:

$$\beta_j(p,q) = P(w_{p,q} | N^j_{p,q}, G)$$

A probabilidade *forward* corresponde à probabilidade de tudo que está fora do domínio de um nodo na árvore, inclusive o próprio nodo, e, da mesma forma que a probabilidade *backward*, pode ser mapeada como a probabilidade *outside* $\alpha_j(p,q)$ no contexto de PCFGs:

$$\alpha_j(p,q) = P(w_{1,p-1}, N^j_{p,q}, w_{q+1,m} | G)$$

A probabilidade *inside* é a probabilidade total de gerar-se as palavras $w_{p,q}$, dado que se está partindo do não terminal N^j. A probabilidade *outside* é a probabilidade total de começar-se com o símbolo inicial N^1 e gerar-se o não terminal $N^j_{p,q}$ e todas as palavras fora de $w_{p,q}$.

[24] Do inglês *Probabilistic Regular Grammar* - PRG.

5.3.2 Questões fundamentais referentes a PCFGs

Como para os HMMs, há também três questões básicas sobre PCFGs que deseja-se responder:

1. Qual a probabilidade de uma sentença $w_{1,m}$, conforme uma gramática G: $P(w_{1,m}|G)$?
2. Qual a derivação mais provável para uma sentença: $\arg\max_t P(t|w_{1,m}, G)$?
3. Como escolher as probabilidades das regras para a gramática G que maximizem a probabilidade da sentença: $\arg\max_G P(w_{1,m}|G)$?

A primeira questão se refere a encontrar a probabilidade de uma determinada sentença. A segunda questão visa decidir qual das derivações de uma sentença é a mais provável, dadas as possíveis seqüências de regras para geração da sentença. A terceira questão mostra a necessidade de treinamento da gramática, para que as probabilidades das regras sejam ajustadas, a fim de melhor representar o corpus utilizado.

Nas subseções a seguir, serão mostradas técnicas para encontrar a probabilidade de uma sentença, para encontrar a melhor seqüência de regras para derivação de uma sentença, e para treinar uma PCFG.

5.3.2.1 A probabilidade de uma sentença

Em geral, não se pode calcular a probabilidade de um sentença simplesmente somando-se as probabilidades das diferentes árvores de derivações possíveis para a sentença, pois este número de derivações cresce exponencialmente. Para calcular-se a probabilidade de uma sentença, pode-se utilizar as probabilidades *inside* ou as probabilidades *outside*.

Usando probabilidades *inside*. Um modo eficiente de calcular-se a probabilidade total de uma sentença é através do algoritmo *inside*. Este é um algoritmo de programação dinâmica, baseado nas probabilidades *inside*.

$$P(w_{1,m}|G) = P(N^1 \overset{*}{\Longrightarrow} w_{1,m}|G) = P(w_{1,m}|N^1_{1,m}, G) = \beta_1(1, m)$$

A probabilidade *inside* de uma subseqüência de palavras é calculada por indução, em relação ao tamanho da subseqüência. No caso básico, quer-se encontrar $\beta_j(k, k)$, probabilidade da regra $N^j \rightarrow w_k$ (em uma gramática na forma normal de Chomsky, há apenas uma maneira possível de um não terminal N^j poder dominar um terminal w_k, que é através da regra $N^j \rightarrow w_k$):

$$\beta_j(k, k) = P(w_k|N^j_{k,k}, G) = P(N^j \rightarrow w_k|G)$$

Em seguida, na indução, quer-se encontrar $\beta_j(p, q)$, para $p < q$. Isto considera todas as maneiras possíveis para derivar-se os terminais dominados por $N^j_{p,q}$. A primeira regra da gramática deve ter a forma $N^j \rightarrow N^r N^s$ (pois considera-se a gramática na forma normal de Chomsky), para poder-se proceder pela indução, dividindo a seqüência em duas em vários pontos da árvore, e somando-se o resultado. Então, $\forall j, 1 \leq p < q \leq m$:

$$\beta_j(p, q) = P(w_{p,q}|N^j_{p,q}, G) = \sum_{r,s} \sum_{d=p}^{q-1} P(N^j \rightarrow N^r\ N^s)\beta_r(p, d)\beta_s(d + 1, q)$$

Utilizando esta relação de recorrência, as probabilidades *inside* podem ser calculadas no sentido *bottom-up* eficientemente.

Usando probabilidades *outside*. Também pode-se calcular a probabilidade de uma sentença através das probabilidades *outside*. Para qualquer k, $1 \leq k \leq m$, tem-se:

$$P(w_{1,m}|G) = \sum_j P(w_{1,k-1}, w_k, w_{k+1,m}, N^j_{k,k}|G) = \sum_j \alpha_j(k,k) P(N^j \rightarrow w_k)$$

As probabilidades *outside* são calculadas no sentido *top-down*, por indução, através do algoritmo *outside*. No caso básico, calcula-se a probabilidade de a raiz da árvore ser um não terminal N^i que não tenha nada fora dele:

$$\alpha_1(1,m) = 1$$
$$\alpha_j(1,m) = 0 \quad \text{para } j \neq 1$$

Em seguida, na indução, em termos do passo anterior da derivação, o nodo $N^j_{p,q}$ atual deve estar a esquerda do nodo antecessor na árvore (assim, o nodo da direita seria $N^g_{q+1,e}$) ou a direita (assim, o nodo da esquerda seria $N^g_{e,p-1}$). Soma-se as duas possibilidades, mas restringe-se, na primeira soma, que $g \neq j$, para que não se conte duas vezes quando as regras forem da forma $X \rightarrow N^j\ N^j$:

$$\begin{aligned}
\alpha_j(p,q) &= \left[\sum_{f,g \neq j} \sum_{e=q+1}^{m} P(w_{1,p-1}, w_{q+1,m}, N^f_{p,e}, N^j_{p,q}, N^g_{q+1,e}) \right] \\
&\quad + \left[\sum_{f,g} \sum_{e=1}^{p-1} P(w_{1,p-1}, w_{q+1,m}, N^f_{e,q}, N^g_{e,p-1}, N^j_{p,q}) \right] \\
&= \left[\sum_{f,g \neq j} \sum_{e=q+1}^{m} \alpha_f(p,e) P(N^f \rightarrow N^j\ N^g) \beta_g(q+1,e) \right] \\
&\quad + \left[\sum_{f,g} \sum_{e=1}^{m} \alpha_f(e,q) P(N^f \rightarrow N^g\ N^j) \beta_g(e,p-1) \right]
\end{aligned}$$

Como com os HMMs, pode-se combinar as probabilidades *inside* e *outside* :

$$\alpha_j(p,q)\beta_j(p,q) = P(w_{1,p-1}, N^j_{p,q}, w_{q+1,m}|G) P(w_{p,q}|N^j_{p,q}, G) = P(w_{1,m}, N^j_{p,q}|G)$$

A probabilidade de uma sentença e da existência de um nodo que engloba as posições de p até q da sentença é dada por:

$$P(w_{1,m}, N_{p,q}|G) = \sum_j \alpha_j(p,q)\beta_j(p,q)$$

5.3.2.2 Encontrando a árvore de derivação mais provável para uma seqüência

Pode-se construir um algoritmo para encontrar a árvore de derivação mais provável para uma sentença através de uma adaptação do algoritmo *inside*, a fim de encontrar-se o elemento que tem a maior soma de probabilidades, e registrar qual regra gerou este valor máximo.

O princípio do algoritmo de Viterbi para os HMMs é definir acumuladores $\delta_j(t)$, que armazenam a maior probabilidade de uma caminho da grade que chega ao estado j no tempo t.

Novamente serão utilizadas as PRGs, para relacionar os HMMs as PCFGs, a fim de encontrar a maior probabilidade de uma árvore de derivação parcial que engloba uma certa subseqüência e que tem como raiz um certo não terminal. Para isso, serão também utilizados acumuladores: $\delta_i(p,q) =$ maior probabilidade *inside* da subárvore $N^i_{p,q}$.

Usando programação dinâmica, pode-se calcular a seqüência de regras de derivação de uma sentença da seguinte forma: em um passo de inicialização, associa-se a cada produção unária em um nodo folha sua

probabilidade. Em um passo de indução, encontra-se a regra binária mais provável, que será a primeira regra, armazenando-a nas variáveis ψ, cujos valores são uma lista de três inteiros que registram a forma de aplicação da regra.

1. Inicialização: $\delta_i(p,q) = P(N^i \rightarrow w_p)$;
2. Indução: $\delta_i(p,q) = \max_{\substack{1 \leq j,k \leq n \\ p \leq r < q}} P(N^i \rightarrow N^j N^k)\delta_j(p,r)\delta_k(r+1,q)$, e $\psi_i(p,q) = \arg\max_{j,k,r} P(N^i \rightarrow N^j\ N^k)\delta_j(p,r)\delta_k(r+1,q)$;
3. Término e leitura do caminho (por *backtracking*): como a gramática tem o símbolo inicial N^i, então a probabilidade da árvore de derivação mais provável cuja raiz é o símbolo inicial é $P(\widehat{t}) = \delta_1(1,m)$.

Quer-se reconstruir $\widehat{t}$, a árvore com a máxima probabilidade. Para isso, deve-se considerar $\widehat{t}$ como um conjunto de nodos $\{\widehat{X}_x\}$, construído da seguinte forma:

- ◇ já que a gramática tem um símbolo inicial, o nodo raiz da árvore deve ser $N^1_{1,m}$;
- ◇ constroem-se os nodos filhos da direita e da esquerda de um nodo não terminal. Se $X_x = N^i_{p,q}$ é o nodo atual e $\psi_i(p,q) = (j,k,r)$, então:
$$\text{esquerda}(\widehat{X}_x) = N^j_{p,r}$$
$$\text{direita}(\widehat{X}_x) = N^k_{r+1,q}$$
- ◇ aplica-se o processo recursivamente, até chegar-se aos nodos folha.

5.3.2.3 Treinando uma PCFG

O treinamento de uma PCFG é um processo que tenta encontrar as melhores probabilidades para as regras da gramática. Como no treinamento de HMMs, é necessário um corpus de treinamento.

De acordo com [JUR00], há duas maneiras de associar-se probabilidades às regras de uma gramática, conforme o corpus de treinamento.

A maneira mais simples ocorre quando as sentenças do corpus já estão sintaticamente analisadas, ou seja, já se tem as árvores de derivação das sentenças. Tal tipo de corpus corresponde a um *treebank*. Dado um *treebank*, a probabilidade de expansão de cada não terminal pode ser computada pela contagem do número de vezes que a expansão ocorre em relação ao número total de vezes que o não terminal aparece no lado esquerdo de uma regra.

Quando não se tem um *treebank* à disposição, precisa-se estimar as probabilidades das regras. Para isso, como no caso dos HMMs, é utilizado um algoritmo de maximização da expectativa, o algoritmo *inside-outside*, que é uma generalização do algoritmo *forward-backward*. O algoritmo *inside-outside* permite ajustar os parâmetros de uma PCFG a partir de sentenças da linguagem não marcadas.

Supõe-se que a uma boa gramática é aquela em que as sentenças do corpus de treinamento são mais prováveis de ocorrer. Desta forma, procura-se a gramática que maximiza a probabilidade dos dados de treinamento.

Primeiramente, será mostrado o treinamento a partir de uma sentença isolada e, então, o processo será estendido para o treinamento de um corpus, assumindo-se a independência entre sentenças.

Para determinar a probabilidade das regras, deseja-se calcular:

$$\widehat{P}(N^j \rightarrow \zeta) = \frac{C(N^j \rightarrow \zeta)}{\sum_\gamma C(N^j \rightarrow \gamma)}$$

onde $C(\cdot)$ é a contagem do número de vezes que uma determinada regra é usada.

Como não se conhecem as probabilidades reais das regras, não se pode calcular as freqüências relativas. Ao invés disso, utiliza-se um algoritmo iterativo para improvisar estimativas. Inicia-se com uma topologia

inicial para a gramática, que especifica quantos terminais e não terminais são utilizados, e algumas estimativas iniciais para as regras. Utiliza-se a probabilidade de cada árvore de derivação de uma sentença de treinamento de acordo com a gramática, acreditando-se nas probabilidades estimadas, e então somam-se as probabilidades de cada regra utilizada em cada ponto da gramática, para ter-se uma expectativa da freqüência com que cada regra é usada. Estas expectativas são usadas para refinar as estimativas das probabilidades das regras; assim, a probabilidade do corpus de treinamento, em relação à gramática, aumenta.

Considera-se:

$$\alpha_j(p,q)\beta_j(p,q) = P(N^1 \stackrel{*}{\Longrightarrow} w_{1,m}|G)P(N^j \stackrel{*}{\Longrightarrow} w_{p,q}|N^1 \stackrel{*}{\Longrightarrow} w_{1,m}, G)$$

Sabe-se como calcular $P(N^1 \stackrel{*}{\Longrightarrow} w_{1,m})$. Então tem-se:

$$P(N^j \stackrel{*}{\Longrightarrow} w_{p,q}|N^1 \stackrel{*}{\Longrightarrow} w_{1,m}, G) = \frac{\alpha_j(p,q)\beta_j(p,q)}{P(N^1 \stackrel{*}{\Longrightarrow} w_{1,m})}$$

A estimativa de quantas vezes o não terminal N^jé usado na derivação é:

$$E(N^j \text{ é usado na derivacão}) = \sum_{p=1}^{m}\sum_{q=p}^{m}\frac{\alpha_j(p,q)\beta_j(p,q)}{P(N^1 \stackrel{*}{\Longrightarrow} w_{1,m})}$$

Quando não se está tratando de um nodo pré-terminal, ou seja, um nodo não terminal cujo único filho é um nodo terminal, substitui-se a definição indutiva de β na fórmula acima, e então, $\forall r, s, p < q$:

$$P(N^j \to N^r\ N^s \stackrel{*}{\Longrightarrow} w_{p,q}|N^1 \stackrel{*}{\Longrightarrow} w_{1,m}, G) = \frac{\sum_{d=p}^{q-1}\alpha_j(p,q)P(N^j \to N^r\ N^s)\beta_r(p,d)\beta_s(d+1,q)}{P(N^1 \stackrel{*}{\Longrightarrow} w_{1,m})}$$

Então, a estimativa de quantas vezes esta regra particular é usada na derivação pode ser encontrada através das somas de todos os intervalos de palavras que o nodo pode dominar:

$$E(N^j \to N^r\ N^s, N^j \text{ é usado}) = \frac{\sum_{p=1}^{m-1}\sum_{q=p+1}^{m}\sum_{d=p}^{q-1}\alpha_j(p,q)P(N^j \to N^r\ N^s)\beta_r(p,d)\beta_s(d+1,q)}{P(N^1 \stackrel{*}{\Longrightarrow} w_{1,m})}$$

Agora, para o passo de maximização, quer-se:

$$P(N^j \to N^r\ N^s) = \frac{E(N^j \to N^r\ N^s, N^j \text{ é usado})}{E(N^j \text{ é usado})}$$

Assim, a fórmula para reestimação é:

$$\begin{aligned}\widehat{P}(N^j \to N^r\ N^s) &= \frac{E(N^j \to N^r\ N^s, N^j \text{ é usado})}{E(N^j \text{ é usado na derivacão})} \\ &= \frac{\sum_{p=1}^{m-1}\sum_{q=p+1}^{m}\sum_{d=p}^{q-1}\alpha_j(p,q)P(N^j \to N^r\ N^s)\beta_r(p,d)\beta_s(d+1,q)}{\sum_{p=1}^{m}\sum_{q=p}^{m}\alpha_j(p,q)\beta_j(p,q)}\end{aligned}$$

Para nodos pré-terminais, procede-se de forma semelhante:

$$\begin{aligned}P(N^j \to w^k|N^1 \stackrel{*}{\Longrightarrow} w_{1,m}, G) &= \frac{\sum_{h=1}^{m}\alpha_j(h,h)P(N^j \to w_h, w_h = w^k)}{P(N^1 \stackrel{*}{\Longrightarrow} w_{1,m})} \\ &= \frac{\sum_{h=1}^{m}\alpha_j(h,h)P(w_h = w^k)\beta_j(h,h)}{P(N^1 \stackrel{*}{\Longrightarrow} w_{1,m})}\end{aligned}$$

onde $P(w_h = w^k)$ é 0 ou 1. Então:

$$\widehat{P}(N^j \rightarrow w^k) = \frac{\sum_{h=1}^{m} \alpha_j(h,h) P(w_h = w^k) \beta_j(h,h)}{\sum_{p=1}^{m} \sum_{q=p}^{m} \alpha_j(p,q) \beta_j(p,q)}$$

Até agora, consideramos o treinamento a partir de uma única sentença. A seguir, assume-se um conjunto de várias sentenças de treinamento $W = (W_1, ..., W_\omega)$, com $W_1 = w_{i,1} \cdots w_{i,m_i}$. Sejam f_i, g_i e h_i os subtermos comuns nos casos de uso de um não terminal, respectivamente, em um nodo que se ramifica, em um nodo pré-terminal e em qualquer lugar. Calcula-se f_i, g_i e h_i para W_i:

$$f_i(p,q,j,r,s) = \frac{\sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \rightarrow N^r\ N^s) \beta_r(p.d) \beta_s(d+1,q)}{P(N^1 \stackrel{*}{\Longrightarrow} W_i | G)}$$

$$g_i(h,j,k) = \frac{\alpha_j(h,h) P(w_h = w^k) \beta_j(h,h)}{P(N^1 \stackrel{*}{\Longrightarrow} W_i | G)}$$

$$h_i(p,q,j) = \frac{\alpha_j(p,q) \beta_j(p,q)}{P(N^1 \stackrel{*}{\Longrightarrow} W_i | G)}$$

Se assume-se que as sentenças de um corpus de treinamento são independentes, então a probabilidade do corpus de treinamento é simplesmente o produto das probabilidades das suas sentenças, de acordo com a gramática. Assim, no processo de reestimação, pode-se somar as contribuições das múltiplas sentenças, gerando-se as seguintes fórmulas:

$$\widehat{P}(N^j \rightarrow N^r\ N^s) = \frac{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i - 1} \sum_{q=p+1}^{m_i} f_i(p,q,j,r,s)}{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i} \sum_{q=p}^{m_i} h_i(p,q,j)}$$

$$\widehat{P}(N^j \rightarrow w^k) = \frac{\sum_{i=1}^{\omega} \sum_{h=1}^{m_i} g_i(h,j,k)}{\sum_{i=1}^{\omega} \sum_{p=1}^{m_i} \sum_{q=p}^{m_i} h_i(p,q,j)}$$

No algoritmo *inside-outside*, deve-se repetir este processo de reestimação de parâmetros até que a variação da probabilidade estimada do corpus de treinamento seja pequena. Se G_i é a gramática na iteração de número i do algoritmo, então garante-se que a probabilidade do corpus de acordo com o modelo melhorará ou ficará estável: $P(W|G_{i+1}) \geq P(W|G_i)$.

Problemas com o algoritmo *inside-outside*. O algoritmo *inside-outside* para treinamento de PCFGs apresenta alguns problemas.

O algoritmo é muito sensível à inicialização dos parâmetros, podendo encontrar máximos locais com muito mais freqüência que nos HMMs.

Experimentos com linguagens artificiais indicaram que, para um bom aprendizado da gramática, exigem-se muito mais não terminais do que o necessário para descrever um linguagem.

Não há garantia de que os não terminais aprendidos corresponderão aos não terminais normalmente induzidos na análise lingüística (NP, VP, PP, etc.).

5.4 Análise sintática probabilística

O objetivo da análise sintática é, dada uma sentença s, obter as árvores de derivação para s de acordo com uma gramática G. Na análise sintática probabilística, pode-se estabelecer um *ranking* entre as possíveis análises, mostrando quão provável cada uma é ou, então, pode-se retornar apenas a análise mais provável para a sentença dada.

A análise sintática pode ser considerada como uma implementação direta da idéia de *chunking*, ou seja, reconhecer unidades estruturais de alto nível que permitam compactar a descrição de uma sentença.

Um analisador sintático é um sistema que estabelece uma estrutura sobre uma sentença arbitrária. Um analisador sintático probabilístico pode ser usado para escolher entre diferentes análises de uma sentença ambígua a mais provável, pois utiliza uma gramática probabilística. Dentre as gramáticas probabilísticas existentes, são utilizadas as PCFGs, mostradas na seção anterior. As PCFGs associam a mais alta probabilidade às construções da linguagem mais freqüentes. Desta forma, sua principal contribuição é a resolução da ambigüidade sintática de sentenças.

Um ponto importante da análise sintática é o processo de aprendizado de gramáticas. Dá-se a uma ferramenta de aprendizado alguns exemplos dos tipos de árvores de derivação que deseja-se reconhecer. Uma coleção de exemplos de árvores de derivação é chamada *treebank*. Assim, para determinar-se uma PCFG a partir de um *treebank*, basta calcular-se as freqüências das árvores locais dos exemplos, e então normalizá-las para obter-se as probabilidades. Muitos pesquisadores argumentam que é melhor construir-se *treebank*s do que gramáticas, pois é mais fácil obter-se a análise sintática correta de uma sentença real do que tentar-se determinar quais são todos os possíveis comportamentos de uma regra.

As PCFGs apresentam alguns problemas, relacionados a sua suposição de independência vinda das gramáticas livres de contexto - por definição, uma gramática livre de contexto assume que a expansão de um não terminal é independente da expansão de qualquer outro não terminal. A seguir serão descritos alguns problemas das PCFGs.

5.4.1 Problemas das PCFGs

Em uma PCFG, assume-se que cada regra é independente das demais regras. Isto impõe a ausência de lexicalização, que é importante, por exemplo, para a escolha da posição em uma árvore de derivação a que uma expressão será ligada. Como descrito na seção 4.4.2, o conteúdo léxico das expressões geralmente provê informação suficiente para escolher-se a ligação correta. Para tratar-se esta deficiência, a maneira mais comum e direta de lexicalizar-se uma PCFG é marcar cada nodo de um sintagma com a palavra principal (núcleo) deste. A idéia central deste modelo de lexicalização é que as dependências lexicais mais fortes estão entre os núcleos dos sintagmas e seus dependentes, por exemplo, entre um substantivo (núcleo) e um adjetivo (modificador), ou entre um verbo (núcleo) e um objeto (complemento). Esta dependência é normalmente verdadeira e, portanto, uma estratégia efetiva. No entanto, existem dependências entre pares de palavras em que nenhuma é núcleo do sintagma.

Outro problema das PCFGs está relacionado a idéia de que as probabilidades são livres de contexto. Por exemplo, a probabilidade de um sintagma nominal ser expandido de uma certa maneira é independente de onde ele está na árvore. Esta suposição gramátical é incorreta, pois a probabilidade de expansão de um nodo NP na posição de sujeito varia significativamente em relação a probabilidade de expandí-lo na posição de objeto - por exemplo, pronomes e nomes próprios aparecem mais comumente na posição de sujeito, enquanto NPs que contenham substantivos seguidos de modificadores ocorrem com maior freqüência na posição de objeto.

6 Aplicações

Neste capítulo, serão apresentados brevemente alguns dos principais campos de aplicação do processamento estatístico da linguagem natural. Serão introduzidas as áreas de recuperação e extração de informações, classificação de textos e tradução automática.

Como não serão aprofundados aspectos sobre os métodos sugeridos em cada uma das áreas, maiores detalhes podem ser encontrados em:

- [MAN99], sobre recuperação de informações e classificação de textos;

- ◇ [MAN99] e [JUR00], sobre tradução automática.

Nestas referências, também são indicadas outras fontes de consulta sobre as respectivas áreas de aplicação do PLN.

6.1 Recuperação de informações

A pesquisa em recuperação de informações (IR) visa desenvolver algoritmos e modelos para recuperar informações a partir de repositórios de documentos. Com o ressurgimento de métodos quantitativos em PLN, as conexões com o campo de IR aumentaram.

Um problema clássico em IR é a recuperação ad-hoc, em que o usuário realiza uma consulta, descrevendo as informações desejadas. Como resposta à consulta do usuário, o sistema retorna uma lista de documentos. O problema está na relevância dos documentos fornecidos ao usuário, pelo sistema de IR, como resposta a sua consulta. Se o modelo de recuperação do sistema consistir simplesmente em recuperar documentos que contenham as palavras-chaves especificadas na consulta do usuário, podem ser retornados vários documentos que não pertençam ao domínio de interesse do usuário.

Então, para que as informações recuperadas estejam mais próximas das desejadas pelo usuário, algumas subáreas da IR contam com um corpus de treinamento formado por documentos que estão classificados como relevantes ou irrelevantes para determinadas consultas. Na subárea que trata de classificação de textos, tenta-se associar documentos a duas ou mais categorias predefinidas.

A maioria dos sistemas de IR contém um índice invertido, que é uma estrutura de dados que lista, para cada palavra da coleção de documentos, os documentos que a contém e a freqüência de ocorrência dessa palavra em cada documento. Um índice invertido facilita a procura por documentos que contenham determinadas palavras. Os índices invertidos podem também conter informação sobre as posições das palavras nos documentos. Esta informação adicional, sobre a posição da palavra, possibilita a procura por expressões compostas por mais de uma palavra no texto. Para cada palavra da expressão, faz-se a busca no índice pelos documentos em que esta ocorre. Só são retornados os documentos em que ocorrem todas as palavras da expressão em posições consecutivas. A identificação de expressões em um documento é muito similar ao problema de descoberta de colocações, podendo-se aplicar a esta tarefa várias das técnicas para procura de colocações.

Mas nem todas as palavras são representadas no índice invertido. Palavras funcionais, que não são úteis para busca (artigos, conjunções, alguns verbos auxiliares, etc.) são mantidas em uma lista separada, chamada *stop list*. Com isso, reduz-se o tamanho do índice.

Outra característica comum nos sistemas de IR é o *stemming* (visto na seção 3.2.3), que se refere a uma forma simplificada de análise morfológica, em que só o radical das palavras é utilizado.

A seguir, serão apresentados brevemente tópicos que relacionam o PLN e a IR, como o modelo de espaço vetorial e indexação semântica latente, modelos probabilísticos para distribuição de termos em documentos, e técnicas para segmentação do discurso.

6.1.1 Tópicos em recuperação de informações

Um dos modelos mais usados para IR ad-hoc é o modelo do espaço vetorial. Neste modelo, os documentos e consultas são representados em um espaço multi-dimensional, em que cada dimensão corresponde a uma palavra da coleção de documentos. Os documentos mais relevantes para uma consulta são aqueles representados pelos vetores mais próximos da consulta, isto é, documentos que usam palavras similares às da consulta. Esta proximidade é freqüentemente calculada em relação aos ângulos - escolhe-se os documentos que formem o menor ângulo com o vetor da consulta.

Outro modelo utilizado em IR é o modelo de distribuição de termos. Isto é, quer-se estimar $P_i(k)$, a proporção de vezes que a palavra w_i aparece k vezes em um documento. Modelos de distribuição de termos obtêm as regularidades de ocorrência de palavras em subunidades de um corpus. A maioria destes modelos tenta

caracterizar o quão informativa uma palavra é. Há vários modelos que formalizam a noção de quantidade de informação, uns baseados na distribuição de Poisson, outros na freqüência inversa de documentos, e ainda outros, baseados na freqüência inversa residual de documentos.

Uma fonte diferente de informação sobre termos que pode ser explorada em IR é a co-ocorrência: o fato de que dois ou mais termos ocorrem no mesmo documento várias vezes. O modelo de indexação semântica latente é uma técnica que projeta consultas e documentos em um espaço com dimensões semânticas latentes. Os termos co-ocorrentes são projetados nas mesmas dimensões, e os termos não co-ocorrentes são projetados em dimensões diferentes. Este modelo é a aplicação de técnicas matemáticas específicas, chamadas *singular value decomposition* (SVD), para construção de uma matriz em que uma dimensão é constituída por palavras, e a outra, por documentos.

O tópico da IR relacionado à segmentação do discurso se refere a divisão dos documentos em partes cujos tópicos são coerentes. Isto deve-se à crescente heterogeneidade das coleções de textos e à grande diferença de tamanho entre os documentos. Um algoritmo para particionamento dos textos é o algoritmo *TextTiling*, que procura por partes de um texto onde o vocabulário muda de um assunto para outro. Estes pontos são considerados os limites das unidades de texto. O algoritmo *TextTiling* divide o texto em pequenas unidades de tamanho fixo, denominadas seqüências de *tokens*. O ponto entre cada seqüência é denominado *gap*. Há três principais componentes do algoritmo: o medidor de coesão, o medidor de profundidade e o seletor de limites. O medidor de coesão mede a quantidade de "continuidade do assunto" ou coesão em cada *gap*, isto é, verifica se o mesmo assunto é predominante nas seqüências de *tokens* em ambos os lados do *gap*. O medidor de profundidade associa uma medida de profundidade a cada *gap*, de acordo com sua medida de coesão em relação à dos demais *gaps* que estão em torno deste. Se a coesão dos outros *gaps* é maior que a do *gap* analisado, sua medida de profundidade será alta. Se a coesão do *gap* em questão for similar à dos *gaps* vizinhos, sua medida de profundidade será baixa. O módulo selecionador de limites procura pelos *gaps* com maior medida de profundidade e os seleciona como melhores pontos para segmentação do texto.

Em [REY99], são apresentados diferentes modelos estatísticos para segmentação de textos e são identificados os usos dos limites encontrados.

6.2 Classificação de textos

Uma tarefa de classificação consiste em associar objetos de um universo a duas ou mais classes. Muitos dos processos vistos anteriormente, como marcação de corpora, resolução da ambigüidade semântica e ligação de expressões preposicionais, são tarefas de classificação. Para marcação e resolução da ambigüidade, procura-se uma palavra no contexto e classifica-se esta palavra como sendo uma instância de uma categoria sintática ou de um sentido de seus sentidos. Na tarefa de ligação de expressões preposicionais, as duas classes são as duas possibilidades de ligação. Outras tarefas de classificação em PLN são a identificação do autor de um texto e a identificação da língua de um texto.

Ainda, uma importante tarefa de classificação é a classificação de textos conforme o assunto ou tema do documento. Uma aplicação da classificação de textos é filtrar um conjunto de textos para um grupo de interesse particular.

Em geral, o problema de classificação estatística pode ser caracterizado por um conjunto de treinamento que contém objetos rotulados com uma ou mais classes, uma classe de modelos e um método de treinamento. A classe de modelos é uma família parametrizada de classificadores, e o método de treinamento seleciona um classificador desta família. Os métodos de treinamento podem ser vistos como algortimos para adequação de funções, que procuram por um bom conjunto de valores dos parâmetros. Treinado o classificador, pode-se avaliar a adequação dos parâmetros utilizando-se dados de teste, isto é, dados não usados no treinamento.

Há diferentes técnicas propostas para a tarefa de classificação, entre elas estão: árvores de decisão, modelagem da entropia máxima, redes neurais e a classificação pelos k vizinhos mais próximos[25].

[25] Do inglês *k nearest neighbor classification.*

6.2.1 Técnicas para classificação de textos

Uma árvore de decisão decide se um documento pertence a uma classe c. A árvore é construída a partir de propriedades dos dados de treinamento - considera-se um critério para ramificação dos nodos e um critério de parada, que determina os nodos folhas. Construída a árvore, cada nodo contém: o número de documentos do conjunto de treinamento que pertencem ao nodo, a probabilidade de um membro do nodo estar na classe c, o parâmetro testado no nodo, e o valor deste parâmetro. Ajusta-se a árvore através de uma técnica de *punning* (poda). Classifica-se um documento da seguinte forma: inicia-se no nodo raiz da árvore, testa-se o valor do parâmetro e segue-se ao nodo apropriado, conforme resultado do teste do parâmetro. Repete-se o processo até chegar-se a um nodo folha.

A técnica de modelagem da entropia máxima integra informações provenientes de várias fontes heterogêneas para classificação. Os dados para classificação são descritos por vários atributos. Estes atributos podem ser complexos, e permitem a utilização de conhecimento prévio sobre que tipos de informação espera-se que sejam importantes para a classificação. Cada atributo corresponde a uma restrição no modelo. Dado um conjunto de atributos, computa-se a expectativa de cada atributo com base nos dados de treinamento. Cada atributo define a restrição de que esta expectativa deve ser a expectativa no modelo de entropia máxima. De todas as distribuições de probabilidade que obedecem as restrições, encontra-se a distribuição da entropia máxima.

Um importante família de técnicas de classificação são as redes neurais. O exemplo mais simples de redes neurais são os *perceptrons*. O *perceptron* é um algoritmo iterativo para aprendizado. Os documentos são representados como vetores. O objetivo é aprender um vetor $\overrightarrow{w}$ e um limite θ, tal que a comparação de θ com o resultado do produto escalar entre $\overrightarrow{w}$ e o vetor de documento proveja a decisão de classificação. Decide-se que o documento pertence à classe c se o resultado do produto dos vetores for maior que θ. A idéia básica do algoritmo de aprendizado, a cada iteração, é corrigir o vetor $\overrightarrow{w}$ e θ, a fim de aproximá-los de um critério de adequação.

O princípio básico da regra de classificação pelo vizinho mais próximo é: para classificar um novo objeto, encontra-se o objeto mais semelhante nos dados de treinamento. Uma generalização da regra do vizinho mais próximo é a classificação pelos k vizinhos mais próximos (classificação KNN): ao invés de utilizar-se apenas um vizinho como base para a decisão, consultam-se k vizinhos. A complexidade da KNN está em encontrar uma boa medida de similaridade. Escolhida a medida de similaridade, decide-se que a classe do novo documento é classe na maioria dos k vizinhos semelhantes.

Em [YAN99], é apresentada uma avaliação de abordagens estatísticas para classificação de textos.

6.3 Tradução automática

A tradução automática de textos de uma língua para outra é uma das mais importantes aplicações de PLN. Há diferentes abordagens para realização desta tarefa.

A abordagem mais simples é a tradução palavra por palavra, onde o problema óbvio é que não há correspondência “um para um” entre as palavras nas diferentes línguas (uma razão para isso é a ambigüidade léxica). Além disso, a ordem das palavras também se altera de uma língua pra outra. Este problema é tratado pela abordagem de transferência sintática.

A abordagem de transferência sintática, primeiramente, analisa sintaticamente o texto na língua original e, então, transforma a árvore de derivação do texto original em uma árvore sintática na outra língua, utilizando regras apropriadas. Finalmente, gera-se a tradução a partir da árvore sintática. Nesta abordagem, um problema é a ambigüidade sintática, pois assume-se que se pode resolver corretamente as ambigüidades do texto original. Além deste, há o problema de que, freqüentemente, uma tradução sintaticamente correta tem uma semântica inapropriada. Por exemplo, em alemão diz-se *Ich esse gern*, que tem uma estrutura verbo-advérbio, e em português, a melhor tradução é *Eu gosto de comer*, que não tem a mesma estrutura - pela abordagem sintática, a frase em alemão seria traduzida, por exemplo, como *Eu como com prazer*.

Na abordagem de transferência semântica, representa-se o significado da sentença original e, então, gera-se a tradução a partir do significado. Esta abordagem resolve a deficiência da associação sintática, mas ainda não é suficiente para todos os casos, pois mesmo que uma sentença tenha seu significado literal traduzido corretamente, esta ainda pode não ser inteligível.

Uma abordagem que não considera as traduções literais é a tradução *interlingua*. *Interlingua* é um método para representação do conhecimento que é independente da maneira como as línguas expressam significados. Trata o problema de tradução para um grande número de línguas diferentes. No entanto, há problemas práticos da abordagem de *interlingua* relacionados à dificuladade em desenvolver formalismos eficientes e compreensíveis para representação do conhecimento e a grande quantidade de ambigüidade que precisa ser resolvida para traduzir a linguagem natural em uma linguagem de representação do conhecimento.

A maioria dos sistemas para tradução automática são uma mistura de componentes probabilísticos (analisador sintático probabilístico, tratamento da ambigüidade semântica, etc.) e não probabilísticos, mas há alguns poucos sistemas de tradução que são completamente estatíticos. Um destes é baseado no modelo do canal com ruído (discutido anteriormente na seção 2.2.4). Neste sistema, para traduzir-se da língua L_1 para a língua L_2, inicializa-se um canal com ruído que recebe como entrada s_2, uma sentença na língua L_2, transforma esta sentença em uma sentença s_1 na língua L_1, e envia s_1 para um decodificador. O decodificador, então, determina uma sentença $\widehat{s_2}$, que seja a mais provável tradução de s_1 ($\widehat{s_2}$ não é necessariamente igual a s_2).

Também no contexto de tradução automática, se encontra um processo chamado alinhamento de textos, que não faz parte propriamente da tarefa de tradução; ao invés disso, o alinhamento de textos é utilizado principalmente para geração de recursos léxicos como dicionários bilíngües e gramáticas paralelas, que melhoram a qualidade da tradução automática.

6.3.1 Alinhamento de textos

O alinhamento de textos consiste em encontrar a correspondência entre unidades de um texto em uma língua e unidades de um texto em outra língua. Este processo é realizado sobre os chamados textos paralelos, que são textos com o mesmo conteúdo, mas em línguas diferentes (por exemplo, manuais de equipamentos eletrônicos, documentos oficiais de países com mais de uma língua oficial, descrição de rituais, etc.).

Obtidos os textos paralelos, o primeiro passo é realizar o alinhamento dos parágrafos ou sentenças, ou seja, verificar quais parágrafos ou sentenças em uma língua correspondem a quais parágrafos ou sentenças na outra língua. O maior problema é que, nem sempre, os tradutores traduzem uma sentença de uma língua em exatamentamente uma sentença na outra língua.

Os métodos mais simples para alinhamento de sentenças são aqueles que se baseiam no comprimento (número de palavras) da sentença - assumem que sentenças longas serão traduzidas por sentenças longas, e sentenças curtas serão traduzidas por sentenças curtas. Estes métodos permitem um alinhamento rápido de grandes quantidades de textos.

Outro tipo de métodos de alinhamento são aqueles baseados em posições, pontos de alinhamento. Esses pontos podem ser definidos por palavras semelhantes nas duas línguas, chamadas cognatos, que são de mesma origem, escritas de maneira semelhante e com mesmo significado. Além dos cognatos, nomes próprios também podem servir como pontos de alinhamento.

Um terceiro tipo de métodos de alinhamento se baseiam em informações léxicas para guiar o processo. Os métodos deste tipo utilizam de forma diferente as informações léxicas sobre as palavras. Em geral, são mais robustos que os demais métodos. Estes métodos atuam em nível das palavras, alinhando, por conseqüência, as sentenças.

Para poder-se utilizar os textos alinhados na geração de dicionários bilíngues, são necessários dois passos: realizar o alinhamento das palavras e, então, utilizar algum critério como, por exemplo, a freqüência, a fim de selecionar os pares alinhados para os quais há evidência suficiente para poderem ser incluídos no dicionário.

7 Conclusão

A crescente disponibilização de dados textuais em formato digital incentivou a adoção e o desenvolvimento de modelos estatísticos para processamento da linguagem natural. Estes modelos, essencialmente, baseiam-se em informações quantitativas obtidas a partir de corpora de textos.

Neste trabalho foram apresentados os principais fundamentos do processamento estatístico da linguagem natural. A seguir, serão enfocados os temas vistos neste trabalho e as conclusão alcançadas.

7.1 O que foi visto

Primeiramente, foram enfocadas as teorias que embasam esta abordagem do PLN, nomeadamente a teoria da probabilidade e a teoria da informação.

Em seguida, foram abordadas questões relacionadas aos problemas encontrados quando se manipulam textos reais no PLN. Enfatizaram-se, também, as principais características de um corpus e os tipos de corpora existentes. Um aspecto importante abordado foi a necessidade de utilizar-se partes diferentes de um mesmo corpus para treinamento e teste dos modelos estatísticos, a fim de obter-se resultados mais confiáveis. Se forem utilizados corpora diferentes, um para treinamento e outro para teste, provavelmente os resultados serão piores, devido às diferenças existentes entre os textos. Se for utilizado o mesmo corpus, tanto para treinamento, como para teste, os resultados são extremamente bons, mas não têm validade, visto que o modelo já estava otimizado de acordo com os dados.

Foram apresentadas as principais tarefas do processamento estatístico da linguagem que necessitam de informações obtidas a partir das palavras. Estas tarefas incluem a detecção de colocações, a estimação da probabilidade de elementos da linguagem, a resolução da ambigüidade semântica e a aquisição de conhecimento lexical.

Dentre os métodos estatísticos existentes para detecção de colocações, foram apresentadas abordagens mais simples, como o método da freqüência, e abordagens mais complexas, como os testes de hipótese. Os testes de hipótese são mais confiáveis, por não serem tão influenciados pela esparcidade dos dados. O teste das taxas de probabilidade é considerado o mais fácil de ser interpretado entre os testes de hipóteses, pois é analisada a relação de dependência entre uma palavra e outra.

Diferentes métodos de inferência estatística foram apresentados para estimar diferentes elementos da linguagem. O modelo de *n*-gramas foi apresentado como método para prever-se a palavra seguinte em uma sentença, dadas as palavras anteriores. Os demais métodos apresentados visavam tratar o problema de estimação da probabilidade de palavras desconhecidas, ou seja, não presentes no corpus de treinamento. Dentre estes métodos, foram considerados mais confiáveis aqueles que, além dos dados de treinamento, utilizam dados de *held-out* para validação das estimativas.

Para resolução da ambigüidade semântica de uma sentença, foram apresentadas abordagens supervisionadas, abordagens baseadas em dicionários, e abordagens não supervisionadas. A utilização de uma abordagem ou de outra depende do material disponível para treinamento. O tratamento da ambigüidade semântica é importante em sistemas de tradução automática, em que a palavra correspondente na outra língua pode variar de acordo com o sentido da palavra na língua original do documento; e em sistemas de recuperação de informações, que devem retornar apenas documentos que atendam a determinado sentido de uma palavra ambígua utilizada na consulta.

A aquisição de conhecimento lexical, utilizada para suprir as deficiências dos dicionários existentes, foi abordada sobre quatro aspectos: subcategorização de verbos, resolução da ambigüidade de ligação, preferências selecionais e similaridade semântica. A subcategorização de verbos consiste em determinar quadros de subcategorização para os verbos de acordo com a categoria sintática dos complementos que admitem, a fim de auxiliar a análise sintática de uma sentença, pois pode-se associar corretamente os complementos aos verbos ou a outros elementos da sentença. A resolução da ambigüidade de ligação, que é um problema sintático, pode ser tratada através de informações sobre as propriedades léxicas das palavras, para o que é proposto o

algoritmo de Hindle e Rooth. Considerou-se principalmente o problema da ligação de expressões preposicionais, que é o mais discutido na literatura. A observação das preferências selecionais de um lexema pode ser utilizada para prever-se parte do significado de uma palavra ocorrida no corpus, mas não contida no dicionário. O último aspecto de aquisição léxica, a similaridade semântica, visa determinar o significado de uma palavra nova a partir da obtenção automática de uma medida de similaridade desta palavra em relação às palavras conhecidas.

Foram também apresentados os princípios do processamento estatístico da linguagem para tratamento da sintaxe de uma linguagem natural. Detalhou-se o processo de marcação de um corpus com as categorias das palavras, sendo que este processo visa resolver as ambigüidades sintáticas do corpus, a fim de facilitar o processo seguinte de análise sintática. Como modelo estatístico para a marcação de um corpus foram apresentados os modelos de Markov escondidos, que são capazes de retornar a melhor seqüência de rótulos de categorias das palavras de uma sentença ambígua.

Abordou-se, em seguida, o processo de análise sintática probabilística do corpus, que utiliza como modelo para representação da estrutura sintática da linguagem uma gramática livre de contexto probabilística. Estas gramáticas têm um comportamento semelhante aos modelos de Markov escondidos, sendo também capazes de retornar a melhor análise sintática de um texto ambíguo. No entanto, há algumas limitações das PCFGs, que são: a não utilização de informações léxicas do corpus, e a não consideração da posição do sintagma na árvore para a estimação das probabilidades das regras.

As aplicações de PLN referenciadas foram a recuperação de informações, a classificação de texto e a tradução automática. Estas áreas se beneficiam dos resultados dos processos de tratamento estatístico da linguagem, descritos anteriormente. Por exemplo, um sistema de recuperação de informações, que visa recuperar informações a partir de repositórios de documentos, necessita de um módulo que resolva a ambigüidade semântica das sentenças, a fim de retornar somente documentos de interesse do usuário do sistema. Já um sistema de tradução automática, além de necessitar da resolução da ambigüidade semântica, ainda necessita do tratamento da ambigüidade sintática.

7.2 Conclusões

Com este trabalho, pôde-se reunir os principais fundamentos do processamento estatístico em um documento em português, pois não foram encontradas referências bibliográficas sobre os métodos estatísticos para PLN em português. Sob este aspecto, tentou-se traduzir da melhor forma os termos específicos empregados em inglês, preservando o seu significado real.

Realizou-se uma busca na Internet, em *sites* de diversos grupos de pesquisa em PLN, por trabalhos que utilizassem a abordagem estatística para processamento da língua portuguesa. No entanto, praticamente nenhum trabalho foi encontrado. A maioria dos trabalhos que utilizam métodos estatísticos para PLN tratam a língua inglesa. Isto indica o caráter inovador de uma pesquisa que utilize o enfoque estatístico para tratar qualquer ponto do PLN.

Pretende-se continuar o estudo do processamento estatístico da linguagem natural, aprofundando-se no estudo das aplicações mostradas no último capítulo. Principalmente, objetiva-se estudar técnicas para a extração e recuperação de informações em grandes bases textuais através de métodos estatísticos. Serão investigados quais recursos são necessários e estão disponíveis para o tratamento estatístico de bases de textos, como corpus marcados em português, rotuladores para a língua portuguesa, etc.

Referências

[ALL95] ALLEN, James. **Natural language understanding**. [S.l.]: Benjamin/Cummings Publishing Company, 1995. 654p.

[CHA93] CHARNIAK, Eugene. **Statistical language learning**. [S.l.]: The MIT Press, 1993. 170p.

[CHU93] CHURCH, Kenneth; MERCER, Robert. Introduction to the special issue on computational linguistics using large corpora. **Computational Linguistics**, v.19, n.1, p.1–24, 1993.

[EPS86] EPSTEIN, Isaac. **Teoria da informação**. São Paulo: Ática, 1986. 77p.

[JUR00] JURAFSKY, Daniel; MARTIN, James H. **Speech and language processing**: an introduction to natural language processing, computational linguistics, and speech recognition. New Jersey: Prentice Hall, 2000. 934p.

[KRE97] KRENN, Brigitte; SAMUELSSON, Christer. **The linguist's guide to statistics**. [S.l.: s.n.], 1997. 170p.

[LEE99] LEE, Lilian. Measures of distributional similarity. In: ACL99, 1999, Maryland. **Anais...** 1999. p.25–32.

[MAN99] MANNING, Christopher D.; SCHÜTZE, Hinrich. **Foundations of statistical natural language processing**. Cambridge, Massachusetts: The MIT Press, 1999. 680p.

[MIH99] MIHALCEA, Rada; MOLDOVAN, Dan I. Disambiguation of unrestricted text. In: ACL99, 1999, Maryland. **Anais...** 1999. p.152–158.

[MOR95] MORETTIN, Luiz Gonzaga. **Estatística básica**: probabilidade. São Paulo: McGraw-Hill, 1995. 185p.

[REY99] REYNAR, Jeffrey C. Statistical models for topics segmentation. In: ACL99, 1999, Maryland. **Anais...** 1999. p.357–364.

[THE99] THEDE, Scott; HARPER, Mary P. A second-order hidden Markov model for part-of-speech tagging. In: ACL99, 1999, Maryland. **Anais...** 1999. p.175–182.

[VIL95] VILLAVICENCIO, Aline. **Avaliando um rotulador estatístico de categorias morfo-sintáticas para a língua portuguesa**. Porto Alegre: CPGCC - UFRGS, 1995. Dissertação de Mestrado.

[YAN99] YANG, Yiming. An evaluation of statistical approches to text categorization. **Information Retrieval**, v.1, n.1/2, p.67–88, 1999.