2
O Estado da Arte

Este capítulo apresenta alguns dos recentes trabalhos relacionados à área de mineração de texto. Na literatura da área é possível definir uma taxonomia de modelos de mineração de texto que inclui o booleano (Wartik, S., 1992), o espaço-vetorial (Salton, G. et al, 1997), o probabilístico (van Rijsbergen, C. J., 1992), o difuso (Subasic, P. e Huettner, A., 2001), o da busca direta (Baeza-Yates, B. e Ribeiro Neto, B., 1999), e os lógicos indutivos (Hu, X. R. e Atwell, E., 2003).

A quantidade de trabalhos publicada nessa área cresce cada vez mais, sendo impossível falar sobre todos os tipos de trabalhos realizados. Alguns trabalhos relacionados à mineração de textos, referenciados na literatura, que utilizam diferentes algoritmos de aprendizado podem ser encontrados em (Aas, K. e Eikvil, L., 1999); (Apté, C. et al, 1994); (Cohen, W. W. e Hirsh, H., 1998); (Cohen, W. W. e Singer, Y., 1996); (Joachims, T., 1997); (Krista, L., 2000); (Li, H. e Yamanishi, K., 2002); (Li, Y. e Jain, A., 1998); (Moulinier, I. e Ganascia, J.-G., 1996); (Thomas, J. e Sycara, K., 1999); (Yang, Y. e Liu, X., 1999).

Dessa forma, escolheu-se apenas alguns importantes e distintos trabalhos para fazer uma descrição das abordagens existentes no estado da arte.

2.1.
Modelos Puramente Estatísticos

As abordagens estatísticas têm a característica fundamental de tentar estimar probabilidades para as decisões tomadas, e normalmente fazem uso de visualização espacial dos dados.

Dentro dessa classe de soluções será mostrado o modelo de Espaço-Vetorial através de (Salton, G. et al, 1997), o modelo de análise de correspondências através de (Lebart, L. et al, 1998) e o de análise de discriminantes através de (Aggarwal, C. C. et al, 1999).

2.1.1.
Modelo de Espaço Vetorial

O modelo Espaço Vetorial é uma das técnicas mais usadas em mineração de textos, sendo a aplicação mais comum a classificação automática de documentos. No contexto do tratamento de documentos, o objetivo principal de um modelo de representação é a obtenção de uma descrição adequada da semântica do texto, de uma forma que permita a execução correta da tarefa alvo, de acordo com as necessidades do usuário (Gean, C. C. e Kaestner, C. A. A., 2004).

De acordo com o modelo vetorial de (Salton, G. et al, 1997), cada documento é representado por um vetor no espaço m-dimensional, onde m é o número de diferentes termos presentes na coleção. Os valores das coordenadas do vetor que representa o documento estão associados aos termos, e usualmente são obtidos a partir de uma função relacionada à freqüência dos termos no documento e na coleção.

Formalmente, seja $C = (d_1, d_2, ..., d_n)$ uma coleção qualquer não-ordenada de documentos d_i, contendo m diferentes termos. Então a representação de um documentos será $d_i = (f_{i1}, f_{i2}, \ldots, f_{im})$ para i = 1 até N, onde f_{ij} é uma função de avaliação associada ao termo j no documento i.

Uma função de avaliação (ou "peso") f_{ij} bastante utilizada é a frequência linear das palavras (TFIDF). Cada termo diferente adiciona uma nova dimensão ao problema. Problemas de mineração de textos costumam apresentar dimensões elevadas. Cada documento será então representado pelo mesmo número m de dimensões indicando a ocorrência do termo no texto.

A classificação de documentos pode ser definida sobre o modelo vetorial como um caso especial de um problema de classificação supervisionada no contexto do Reconhecimento de Padrões (Duda, R. O. et al, 2000).

Um classificador bem conhecido na área do Reconhecimento de Padrões é o k-vizinhos mais próximos (k-NN) (Duda, R. O. et al, 2000). Este algoritmo é amplamente utilizado devido à sua simplicidade conceitual e erro conceitualmente limitado. De maneira abreviada, um classificador k-NN associa um documento d à classe mais freqüente entre as classes dos k vizinhos mais próximos de d na coleção, de acordo com uma distância calculada no espaço vetorial de documentos.

Na área do tratamento de textos, a distância entre dois documentos di e dj mais comumente utilizada é a distância euclidiana.

$$(d_i,\ d_j) = \left[\sum_{k=1}^{M}(f_{ik} - f_{jk})^2\right]^{1/2}$$

e a denominada "métrica do co-seno"

$$cos(d_i, d_j) = \frac{d_i * d_j}{\|d_i\| * \|d_j\|}$$

Devido à dimensão elevada do espaço de documentos (M), nessa abordagem divide-se o espaço original em diversos subespaços, cada qual tratado por um classificador específico.

Considere-se o caso de P subespaços: inicialmente algumas colunas da matriz de (documentos x termos) C são selecionadas aleatoriamente. Se 1,2,…M são as colunas de C, seja X o subespaço projeção sobre estas colunas; proj X (C) representa a sub-matriz obtida de C pela projeção de suas linhas sobre X, com dimensão N x | X |, e proj X (d) é a matriz 1 x | X | que corresponde a um documento d.

Em cada subespaço gerado desta forma, um classificador pode atuar. Nos experimentos constantes deste trabalho foram utilizados subespaços de mesma dimensão (isto é | X | é constante para cada subespaço X). Em cada X empregou-se um classificador k-NN fundamentado na métrica do co-seno com o critério usual de classificação do algoritmo. Por exemplo, para k=1 segue-se o seguinte critério de classificação: Classe (d) = Classe(di) onde di é tal que cos(di,d) < cos(dj,q) para todo j <> i.

Quando se aplica a regra de classificação em cada subespaço, obtém-se P possíveis classificações. Então se deve decidir a classe de d usando um procedimento de decisão que leve em conta os resultados individuais dos diferentes classificadores de 1 até P. Usualmente para a combinação de classificadores se emprega o princípio do voto da maioria (*majority vote principle*), isto é, assinala-se ao documento d a classe mais freqüente entre as P assinaladas individualmente pelos classificadores a d.

Além destas regras, (Gean, C. C. e Kaestner, C. A. A., 2004) empregaram uma segunda regra de combinação: inicialmente um conjunto com todos os documentos que se constituem nos vizinhos mais próximos a d é formado; em

seguida determina-se a classe de cada um destes documentos e a mais freqüente é indicada. Este procedimento considera apenas documentos diferentes para calcular a classe final, visto que a formação do conjunto intermediário elimina aparecimentos múltiplos dos documentos, não importando o número de vezes em que os mesmos apareçam nas P classificações.

Em especial, um implementação dessa técnica que vem ganhando bastante visibilidade na literatura é a de Joachims (2002) - Support Vector Machines.

2.1.2. Análise de Correspondência

Esta abordagem provém de uma técnica bastante conhecida em estatística para análise de associações entre palavras. O resultado da análise permite que um ser humano interprete visualmente as associações, enxergue conglomerados e assim extraia um conhecimento relevante. Em mineração de textos esta é uma boa combinação já que o tratamento analítico das palavras ainda é complexo para o computador e o humano, com conhecimento no assunto, pode apurar a análise.

Um dos representantes deste trabalho é o francês Ludovic Lebart. Em seus trabalhos, Lebart se preocupou muito com a aplicação de pesquisa de mercado. Esse tipo de problema envolve um trabalho de campo onde várias pessoas recebem questionários de perguntas abertas e preenchem com texto livre.

Para (Lebart, L. et al, 1998) a idéia é construir uma tabela de contingência lexical com as palavras em linhas e as categorias em coluna formando uma matriz C(i,j), onde o valor (i,j) é o número de ocorrências da palavra j no questionário i.

Como a quantidade de palavras é muito grande, é feito um pré-processamento selecionando as palavras que têm freqüência maior que **fc**. Isto ajuda a reduzir a dimensionalidade do problema e filtrar as associações mais significantes.

Em um exemplo, um questionário com a pergunta “O que é importante para sua vida?” foi entregue a uma população que era caracterizada por educação e idade. Para formar a tabela de contingência lexical, executou-se um corte de freqüência de 16 ou mais vezes e consolidaram-se as variáveis educação e idade em 9 categorias diferentes.

Tabela 2 – Matriz com o número de ocorrência do termo (linhas) em cada uma das partições educação (L,M e H) e idade (-30, -55 e +55).

	L-30	L-55	L+55	M-30	M-55	M+55	H-30	H-55	H+55
I	2	46	92	30	25	19	11	21	2
I'm	2	5	9	3	2	1	0	0	0
a	10	56	66	54	44	19	20	22	7
able	1	9	16	9	7	4	4	5	0
about	0	3	13	7	1	2	4	1	0
after	1	8	11	3	1	2	0	0	0
all	1	24	19	8	18	6	3	5	2
and	8	89	148	86	73	30	25	32	13
anything	0	4	9	1	3	0	1	1	0

Como um procedimento de análise de correspondência normal, são calculadas distâncias por meio da estatística de Qui-quadrado:

$$\frac{(\text{frequência observada} - \text{frequência esperada})^2}{\text{frequência esperada}} = \frac{(O-E)^2}{E}$$

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Aplicada a análise de correspondência, a distância entre dois pontos i e i' é dada por

$$d_{ii'}^2 = \sum_j (1/c_j (p_{ij}/r_i - p^2_{i'j}/r_{i'}))$$

onde cj é o total da coluna j, ri é o total da linha i e pij o valor da célula.

Os valores das distâncias são organizados em uma matriz de distâncias. É então feita uma visualização dos dados a partir da projeção (de duas dimensões) que maximiza a variância dos dados segundo o procedimento estatístico de análise das componentes principais. O resultado pode ser visto na Figura 3

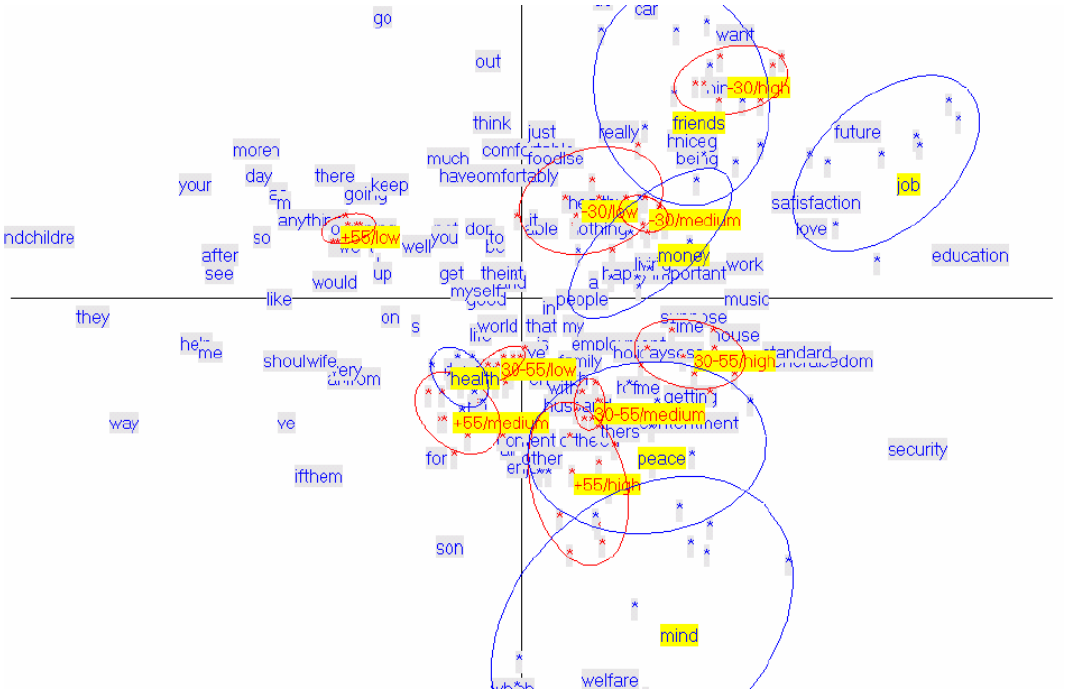


Figura 3 – Resultado em duas dimensões da análise de correspondência

Interpretando este gráfico pode-se notar uma associação interessante entre a palavra “friends” e a categoria “-30/high” indicando que os jovens com alto nível educacional dão bastante importância aos amigos. Já os jovens de baixo e médio nível educacional dão mais importância ao dinheiro (“money”). Os mais velhos e de bom nível educacional (“medium e high”) dão mais importância à paz (“peace”), enquanto os de pouca educação se preocupam com sua saúde (“health”).

Na conclusão, o próprio autor já aponta para a importância de um bom pré-processamento por ter enfrentado dificuldades lexicais no tratamento de elementos redundantes e na delimitação da unidade lexical que corromperam a análise. Para mostrar a importância da etapa de pré-processamento, ele executou esta manualmente através da escolha de alguns segmentos que foram bastante freqüentes e calculou o novo resultado, ilustrado na Figura 4.

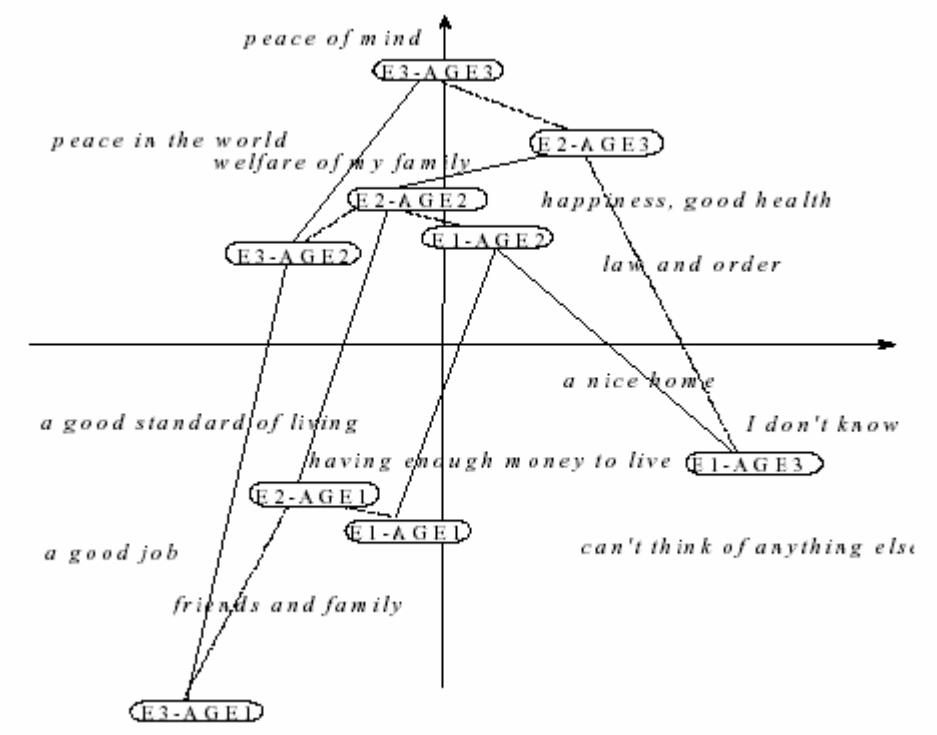


Figura 4 – Análise de correspondência utilizando a etapa de pré com segmentos de frase

Como no experimento da Figura 2, *friends and family* estão próximos a categoria -30/high, *peace of mind* está correlacionado com +55/high, porém *money* não tem mais a ver com -30/*medium* como antes. Isso mostra como um bom pré-processamento dos dados interfere no resultado final e por isso é fundamental em mineração de textos.

2.1.3. Análise de Discriminante

O método estatístico conhecida por análise de discriminante procura achar as palavras que mais discriminam o conjunto de documentos baseado nos conceitos Bayesianos. A diferença para outras abordagens estatísticas é que essa explicita o conhecimento extraído e determinando as palavras e os pesos relativos. Dessa forma, o usuário pode alterar o resultado, adicionando ou excluindo alguma palavra.

No trabalho de (Aggarwal, C. C. et al, 1999), são utilizados índices baseados na freqüência relativa para fazer categorização automática de documentos. Apesar de o processo de categorização também ser feito por

estatísticas em espaço vetorial, o que autor propõe de diferente está no pré-processamento dos dados textuais. O objetivo é reduzir a dimensionalidade do problema e poder executar cálculos em espaço vetorial sem perder performance. Para isso utilizou-se de conhecimentos da teoria de informação para selecionar apenas uma parte das palavras, como descrito a seguir.

Seja K o número de classes distintas, f(K) a freqüência de ocorrência da palavra em cada classe e n(K) o número total de palavras em cada classe. Assim, a freqüência relativa em uma classe particular é definida por f(i)/n(i). Finalmente e proveniente da teoria da informação, foi usado o índice Gini (Breiman, L. et al, 1984), que nesse caso é dado por:

$$G = 1 - \sqrt{\sum_{i=1}^{K} p_i^2}$$

onde

$$p_i^2 = \frac{f_i / n_i}{\sum_{i=1}^{K} f_i / n_i}$$

Se a palavra não for discriminante ela será distribuída igualmente em todas as classes e o índice Gini atinge seu valor máximo $1 - 1/\sqrt{K}$. Por outro lado se a palavra for altamente discriminante então o índice é muito menor.

Outro índice discriminante bastante usado é o TFIDF. A medida *term frequency* – TF – é uma medida que utiliza o número de ocorrências do termo tj no documento di. A idéia é que os termos que mais ocorrem no documento são mais relevantes que os termos menos freqüentes. Nesse caso, é atribuído a aij o valor TF(tj,di), o qual representa o número de vezes que o termo tj ocorre no documento di – Equação 3.2.

$$a_{ij} = TF(t_j, d_i) \quad (3.2)$$

No entanto, um termo muito freqüente também pode ocorrer em quase todo o conjunto de documentos. Quando isso ocorre, esses termos não são úteis para uma boa discriminação das categorias. O componente da coleção é usado para dar

um peso menor para tais termos usando a medida *Inverse Document Frequency – IDF* – definida pela Equação 3.3.

$$IDF = \log \frac{N}{c} \quad (3.3)$$

IDF varia inversamente ao número de documentos *c* que contêm o termo *tj* em um conjunto de documentos *N*. A medida *IDF* favorece termos que aparecem em poucos documentos do conjunto. Assim, as medidas *TF* e *IDF* podem ser combinadas em uma nova medida denominada *TFIDF*. O valor de *aij* pode então ser calculado pela Equação 3.4.

$$a_{ij} = TFIDF(t_j, d_i) = TF(t_j, d_i) \times \log \frac{N}{c} \quad (3.4)$$

O componente de normalização é utilizado principalmente para ajustar os pesos dos atributos para que tanto documentos pequenos quanto documentos maiores possam ser comparados na mesma escala. Em muitas situações, documentos pequenos são representados por poucos termos, enquanto que os documentos maiores, geralmente, por muitos termos. quando uma grande quantidade de termos é usada na representação de documentos, a probabilidade do termo pertencer a um documento é alta e, assim, documentos maiores têm melhores chances de serem relevantes do que documentos menores. Normalmente, todos os documentos relevantes à tarefa deveriam ser tratados com a mesma importância independente do seu tamanho. Um fator de normalização, nesse caso, deve ser incorporado. Os valores *aij* podem então ser formalizados de diversas formas, tais como a definida pela Equação 3.5, aqui denominada de TFIDFN para a medida TFIDF.

$$a_{ij} = TFIDFN(t_j, d_i) = \frac{TFIDF(t_j, d_i)}{\sqrt{\sum_{s=1}^{N} (TFIDF(t_s, d_i))^2}} \quad (3.5)$$

2.2.
Redes Neurais

Os modelos de Redes Neurais foram largamente utilizados durante a década de 90 para diversos fins, inclusive na área de mineração de textos. As Redes Neurais costumam ser modelos complexos, porém fechados, que fornecem bons resultados para determinadas aplicações onde se têm grandes volumes de dados, como é o caso de mineração de textos.

Dentro desta classe de soluções descreveremos três trabalhos: o modelo de Hopfield desenvolvido por (Sergei Ananyan), o modelo de Backpropagation no trabalho de (Fukuda, F., 1999) e redes auto-organizáveis (Kohonen, T. et al, 2000).

2.2.1.
Hopfield

As redes de Hopfield possuem uma das primeiras arquiteturas para redes neurais, e sua importância para a área de mineração de textos se deve à sua retroalimentação.

Sergei Ananyan é da empresa americana Megaputer e ganhou prêmio pelo software de análise de textos TextAnalyst (Ananyan, S., 2006). Este software é baseado em um modelo de redes neurais *Hopfield-like* construído por Ananyan. A seguir descreveremos um pouco do seu funcionamento.

O programa TextAnalyst[2] realiza três processos principais. Primeiro o texto é escaneado para uma variável caracter a caracter. Determina-se então uma janela de 2 a 20 caracteres que é passada pelo texto tirando fotos que serão a representação de palavras. O próximo passo é identificar o quão freqüente as palavras são encontradas juntas em um mesmo trecho semântico. Os parágrafos são contabilizados primeiramente e depois as frases.

Numa segunda etapa o sistema monta uma rede preliminar onde cada palavra e cada relação tem um peso de acordo com a análise de freqüência. Finalmente, essa rede é usada como condição inicial para uma rede neural de Hopfield (Figura 5) com uma dimensão e neurônios totalmente interconectados.

[2] http://www.megaputer.com/ta_algo.html

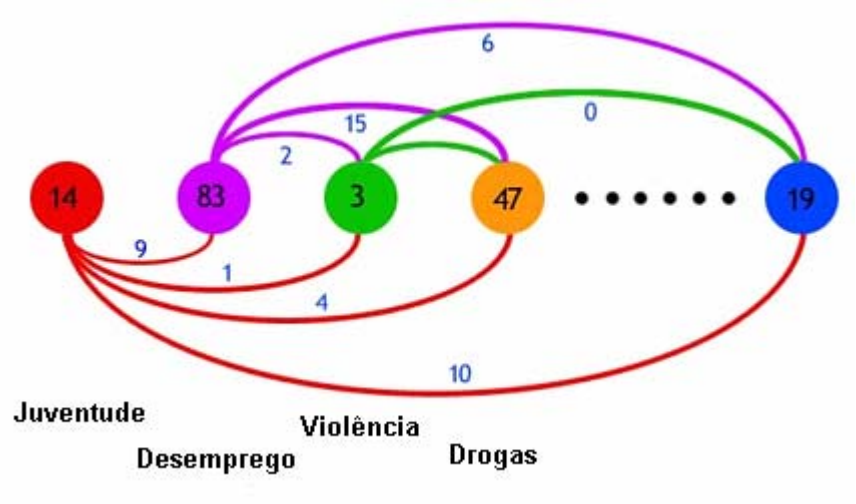


Figura 5 – Arquitetura de Hopfield utilizada no software TextAnalyst

O resultado é uma rede refinada com pesos redefinidos e normalizados, produzindo o que é nomeado de *semantic network*. Essa rede semântica é a base para todas as aplicações como classificação, sumarização e busca.

2.2.2. Backpropagation

A rede neural do tipo *Backpropagation* se tornou a mais utilizada dentre as redes do tipo supervisionado. Uma aplicação deste modelo foi dada por (Fukuda, F., 1999) em sua tese de mestrado na PUC-Rio.

Nesta abordagem supervisionada, o usuário do sistema rotula os textos como positivo e negativo. A partir daí é feito um cálculo de índices que servem de insumo para a entrada da rede neural com uma camada escondida. A saída de treinamento é justamente o rótulo dado ao texto. A arquitetura e o modo de treinamento da rede neural são mostrados na **Figura 6**.

Os índices são calculados da seguinte forma:

```
TP: avaliação dos Termos Positivos
RP: avaliação dos Relacionamentos Positivos
DP: avaliação dos Proximidades Positivas
TN: avaliação dos Termos Negativos
RN: avaliação dos Relacionamentos Negativos
DN: avaliação dos Proximidades Negativos
```

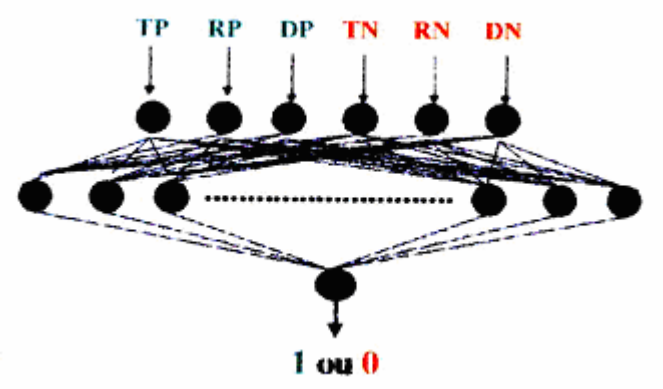


Figura 6 – Rede Neural Backpropagation. Modelo usado por Fukuda.

O cálculo dos índices nada mais é do que um pré-processamento dos dados e é feito com base na freqüência de ocorrência do termo, na freqüência relativa e na proximidade dos termos.

Depois do treinamento, um novo texto pode ser apresentado à rede e ela irá responder 0 ou 1, classificando-o segundo os pesos ajustados durante o treinamento. Os pesos aprendidos guardam o conhecimento da base de treinamento marcada pelo usuário de acordo com seus interesses. Uma quantidade pequena de neurônios é indicada para a captura da essência dos interesses do usuário produzindo uma melhor generalização do conhecimento.

Ainda em (Fukuda, F., 1999), foram apresentadas dificuldades em seu trabalho quanto ao Processamento da Linguagem Natural, não disponível na época para o português. Na bibliografia, observa-se também que a maioria dos trabalhos da literatura de mineração de textos já utilizavam técnicas de PLN, corroborando a importância dessa junção.

2.2.3. Mapas Auto-Organizáveis

Os mapas auto-organizáveis (SOM, na sigla em inglês) são métodos de redes neurais de aprendizado não supervisionado que organizam os dados segundo uma função objetiva. Os neurônios são interligados, dispostos sobre uma condição inicial proveniente dos dados de entrada e, sob treinamento, procuram um equilíbrio de balanceamento da função objetiva.

Em especial, o Kohonen, é um SOM que permite a visualização da similaridade dos dados analisados. Funciona com base na reorganização espacial

dos dados, mantendo a mesma topologia, isto é, documentos semelhantes ficam próximos e documentos diferentes ficam distantes entre si (Kohonen, T. et al, 2000).

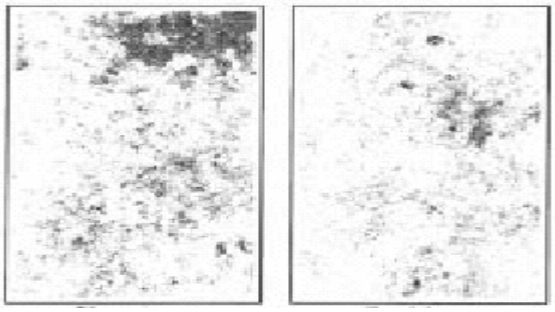
Uma forma de utilizar a arquitetura de Kohonen para classificar textos é descrever cada texto segundo um modelo estatístico de histograma. Podemos ainda dar pesos às palavras e utilizar a IDF (*inverse document frequency*) como peso de informação (Spark-Jones, K. e Willet, P., 1997).

Inicialmente, os neurônios têm pesos aleatórios, mas a cada documento que é apresentado os neurônios da rede competem entre si para saber quem é o vencedor em termos de similaridade, e o vencedor é reajustado, assim como os vizinhos, segundo a seguinte expressão.

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{c(\mathbf{x}),i}(t)\,[\mathbf{x}(t) - \mathbf{m}_i(t)] \qquad (1)$$

$$c(\mathbf{x}) = \arg\min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \qquad (2)$$

Ao final os resultados são visualizados em um mapa. A Figura 7 mostra um exemplo de SOM em equilíbrio. No primeiro mapa foram grifados com pontos escuros os documentos que abordam o textos sobre química, no segundo foram grifados os textos sobre construção. As nuvens pretas mostram os lugares no mapa onde há a maior concentração de documentos das classificações.



Química **Construção**

Figura 7 – Distribuição do mapa dos documentos. Em cinza os lugares onde há a maior concentração de documentos.

Neste experimento, foram visualizados 6840568 documentos mostrando a capacidade de processamento de uma rede neural. Mesmo assim, podemos notar que o pré-processamento dos dados ainda é crítico para o resultado final.

2.3. Aprendizado de Máquina

Dentro da área de aprendizado automático, recentemente tem-se investido muito em Aprendizado de Máquina (ML, do inglês *Machine Learning*) para a resolução de problemas de pré-processamento de textos como: etiquetagem morfossintática (Brill, E., 1995) e (Ratnaparkkhi, A., 1998), identificação de sintagmas nominais básicos (*base noun phrase*) (Cardie, C. e Pierce, D., 1998), (Ramshaw, L. A. e Marcus, M. P., 1995), (Tjong, E. F., 2000); (Tjong, E. F., 2002) e análise sintática parcial (Koeling, R., 2000), (Ramshaw, L. A. e Marcus, M. P., 1995); (Tjong, E. F., 2002). Dentre as técnicas de aprendizado de máquina mais utilizadas podemos citar Cadeias de Markov Escondidas (Seymore, K. et al, 1999), Bayesian Model Merging, Entropia Máxima, Aprendizado Baseado em Casos (*Memory Based Learning*) e Aprendizado Baseado em Transformações (Brill, E., 1993).

Os modelos que serão mencionados nas seções seguintes são baseados em lógica indutiva e regras. Apesar de apresentarem cálculos probabilísticos em sua estrutura, estes não compõem a parte crítica do modelo. São eles: o modelo de Cadeia de Markov através de (Seymore, K. et al, 1999) e o modelo de Aprendizado Baseado em Transformações (TBL, sigla em inglês) através de (Brill, E., 1993).

2.3.1. Cadeias de Markov Escondidas

Este modelo, em inglês Hidden Markov Models (HMM), é extremamente eficiente em seus resultados, porém seu treinamento é bastante custoso.

> "A HMM is a finite state automaton with stochastic transitions and symbol emissions" (Rabiner, L., 1989)

Como já foi dito anteriormente, tem um forte embasamento estatístico, mas as regras envolvidas têm um peso muito maior no resultado.

Um modelo de HMM é composto por:

- Um conjunto de estados escondidos: 1,2,3,...,N
- Um seqüência observada q0 (início), q1, q2, ..., qT, ..., qN (fim)

- Vocabulário de saída: $\Sigma = (\sigma_0, \sigma_1, ..., \sigma m)$
- Probabilidade de Transição P(q → q')
- Verossimilhança da saída B(q ↑ σ)

Seguindo de perto o trabalho de (Seymore, K. et al, 1999), foi utilizado HMM para extração de informação. A informação a ser extraída neste modelo é estrutural, isto é, o objetivo é classificar o trecho de texto quanto a pertinência ao título, ao autor ou ao corpo do documento.

Primeiramente é inserida no sistema o automato de transição dos estados, um exemplo pode ser visto na Figura 8.

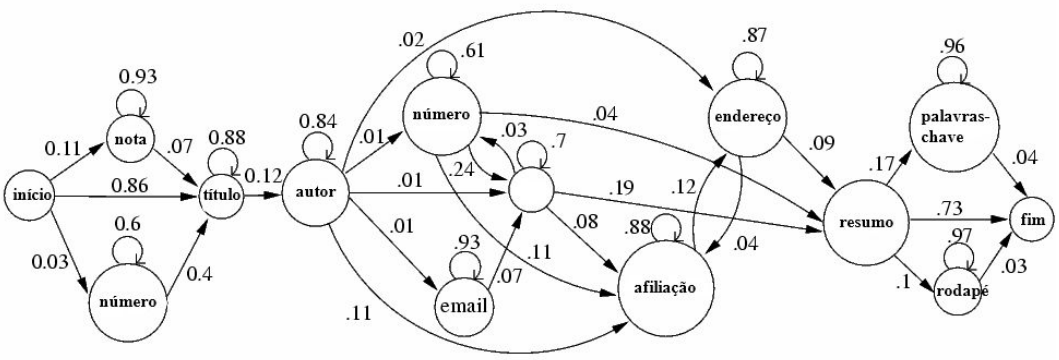


Figura 8 – Exemplo de automato para Cadeia de Markov Escondida.

Uma vez carregado o autômato, é ativado o procedimento de otimização que é calculado a partir da combinação de rótulos que maximiza a verossimilhança. O algoritmo de (Viterbi, A. J., 1967) resolve este problema através da seguinte equação:

$$V(x \mid M) = \arg\max_{q_1 \ldots q_l \in Q^l} \prod_{k=1}^{l+1} P(q_{k-1} \rightarrow q_k) P(q_k \uparrow x_k)$$

O resultado desse procedimento é um conjunto de rótulos para cada palavra do documento. A Figura 9 mostra um texto original a ser pré-processado segundo o automato da Figura 8, as palavras passam por um dicionário indicando a sua classe e o algoritmo de HMM rotula cada uma delas com os rótulos <T> para título, <D> para datas, <A> para autor e <C> para conetúdo, de forma a maximizar a probabilidade de transição. A Figura 10 mostra o texto original rotulado.

Como era esperado, Brasil goleia Hong Kong em amistoso

09:02 09/02

Redação e agências

Apesar do adversário não ser lá muito perigoso, a seleção brasileira fez nesta quarta-feira um bom amistoso contra a seleção de Hong Kong. Para quem queria ver Ronaldinho Gaúcho, Robinho, Roberto Carlos e cia. tocarem fácil na bola até o gol foi uma ótima oportinidade. O placar do 7 a 1 no final nem foi o mais importante.

Figura 9 – Texto original

Como<T> era<T> esperado<T> , <T> Brasil<T> goleia<T> Hong<T> Kong<T> em<T> amistoso<T>

09:02<D> 09/02<D>

Redação<A> e<A> agências<A>

Apesar<C> do<C> adversário<C> não<C> ser<C> lá<C> muito<C> perigoso<C>,<C> a<C> seleção<C> brasileira<C> fez<C> nesta<C> quarta<C>-<C>feira<C> um<C> bom<C> amistoso<C> contra<C> a<C> seleção<C> de<C> Hong<C> Kong<C>.<C> Para<C> quem<C> queria<C> ver<C> Ronaldinho<C> Gaúcho<C>,<C> Robinho<C>,<C> Roberto<C> Carlos<C> e<C> cia<C>.<C> tocarem<C> fácil<C> na<C> bola<C> até<C> o<C> gol<C> foi<C> uma<C> ótima<C> oportinidade<C>.<C> O<C> placar<C> do<C> 7<C> a<C> 1<C> no<C> final<C> nem<C> foi<C> o<C> mais<C> importante<C>.<C>

Figura 10 – Resultado da extração de informação de HMM. Os rótulos após cada palavra é o rótulo ótimo encontrado. <T> título; <D> data; <A> autor e <C> corpo.

2.3.2.
Aprendizado Baseado em Transformações

O aprendizado baseado em transformações é comumente referenciado pela sigla em inglês TBL (*Transformation Based Learning*) e pertence à classe de técnicas automáticas de *machine learning*. Essa técnica está ganhando cada vez mais espaço na literatura devido a seus bons e eficientes resultados.

A saída do TBL é uma lista ordenada de regras, as quais, aplicadas nos dados, fazem reduzir o erro de rotulação. TBL tem sido aplicada em diversas tarefas em mineração de textos como resolução de ambiguidade sintática (Brill, E. e Resnik, P., 1994), parsing sintático (Brill, E., 1993) e desambiguização de palavras (Dini et al, 1998). Mas os melhores resultados dessa técnica têm sido na etiquetagem de classe gramatical. Uma abordagem desta aplicação será descrita nos parágrafos seguintes.

Seguindo o trabalho de (Brill, E., 1995) podemos dizer que um processo genérico de TBL procede como na Figura 11.

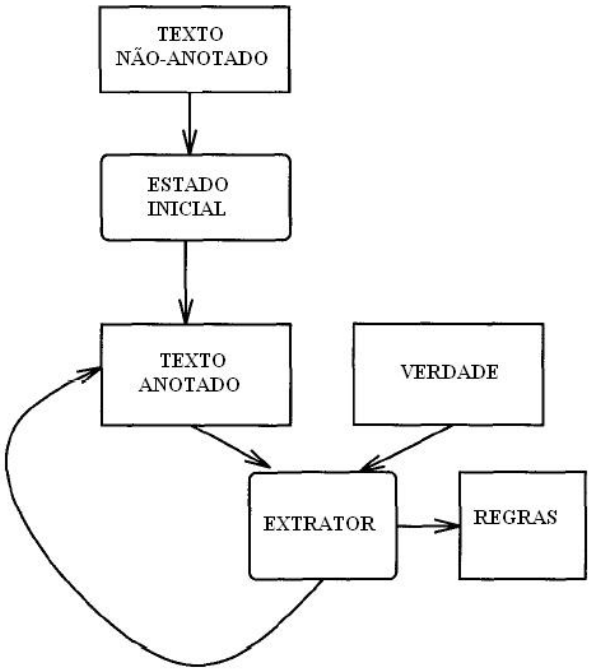


Figura 11 – Processo de funcionamento de um TBL

Primeiro, um texto não anotado é passado por um anotador inicial, que pode ser tanto manual como aleatório. Uma vez rotulado ele é comparado com a verdadeira classificação, que é um corpus de treinamento. Assim, uma lista de

regras de transformação vão sendo aprendidas. Estas regras podem ser de duas formas: as regras de reescrever e a de disparo.

As regras de reescrever agem da seguinte forma: “mude o rótulo de substantivo para adjetivo” e as de disparo das regras de reescrever como “se a palavra anterior é um determinante”.

Os problemas desta abordagem se resumem em precisar de um grande corpus anotado, gerar muitas regras (o que vai de encontro ao princípio da parsimônia) e não contemplar um léxico. O fato de não ter de trabalhar com um léxico impacta na rotulação de uma palavra de um texto novo que não estava contida no corpus anotado.