FERNANDO PEREIRA ALVES DE ARAÚJO

ESTUDO E ADAPTAÇÃO DA TÉCNICA DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO) APLICADA AO AGRUPAMENTO E CLASSIFICAÇÃO EM BASES DE DADOS TEXTUAIS

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS – BRASIL
2008

FERNANDO PEREIRA ALVES DE ARAÚJO

ESTUDO E ADAPTAÇÃO DA TÉCNICA DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO) APLICADA AO AGRUPAMENTO E CLASSIFICAÇÃO EM BASES DE DADOS TEXTUAIS

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de concentração:

Inteligência Computacional

Orientador:

Prof. Dr. Ahmed Ali Abdalla Esmin

LAVRAS
MINAS GERAIS – BRASIL
2008

Ficha Catalográfica preparada pela Divisão de Processo Técnico da Biblioteca Central da UFLA

Araújo, Fernando Pereira Alves

Estudo e Adaptação da Técnica de Otimização por Enxame de Partículas (PSO) Aplicada ao Agrupamento e Classificação em Bases de Dados Textuais / Fernando Pereira Alves de Araújo – Minas Gerais, 2008, 46p.il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Particle Swarm Optimization. 2. Base de dados textual. 3. Classificação 4. Agrupamento I. ARAÚJO, F. P. A.. II. Universidade Federal de Lavras. III. Título.

FERNANDO PEREIRA ALVES DE ARAÚJO

ESTUDO E ADAPTAÇÃO DA TÉCNICA DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO) APLICADA AO AGRUPAMENTO E CLASSIFICAÇÃO EM BASES DE DADOS TEXTUAIS

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em

_____
Prof. Dr. Cláudio Fabiano Motta Toledo

_____
Prof. Msc. Humberto César Brandão de Oliveira

_____
Prof. Dr. Ahmed Ali Abdalla Esmin
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL
2008

Dedico esse trabalho aos meus pais, Agenor e Maria Márcia

AGRADECIMENTOS

Agradeço aos meus pais, Agenor e Maria Márcia, pela educação, motivação e exemplo em minha vida.

Aos meus irmãos, Eduardo e Guilherme, pelo apoio e conselhos.

Aos meus amigos e primeiros orientadores na universidade, Luci e Humberto, que além da amizade me mostraram muitos caminhos na computação, contribuindo para minha formação acadêmica e profissional.

Aos meus amigos do período de graduação, que ajudaram a fazer esses anos de estudo mais prazerosos, entre risadas e cervejas.

Ao orientador Maj. Claret, pela experiência compartilhada, pela amizade e oportunidade de trabalho conjunto.

E aos secretários, Ângela e Deivson, sempre solícitos nas minhas correrias do departamento e ótimas companhias nos intervalos.

ESTUDO E ADAPTAÇÃO DA TÉCNICA DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS (PSO) APLICADA AO AGRUPAMENTO E CLASSIFICAÇÃO EM BASES DE DADOS TEXTUAIS

Buscando aumentar a produtividade e comodidade de indivíduos, além de economia de tempo, a automatização de tarefas tornou-se essencial. Nessa linha de raciocínio, com o aumento das bases de dados pela geração e armazenamento de dados para análise e manuseio futuro, objetivamos nesse trabalho desenvolver e mostrar os resultados de uma técnica de otimização (Otimização por Enxame de Partículas - PSO) com duas novas funções de avaliação aplicada ao problema de agrupamento e classificação automática de bases de dados textuais.

Palavras-chave: otimização por enxame de partículas, PSO, recuperação de informação, classificação, agrupamento, base de dados textual

STUDY AND ADAPTATION OF PARTICLE SWARM OPTIMIZATION TECHNIQUE (PSO) APPLIED TO TEXTUAL DATABASES CLUSTERING AND CLASSIFICATION

In the way of improve individual's productivity and convenience, besides reduction of time spent, task automation becomes essential. With growth of data bases size by encountering information and storing the correspondent data for future analysis and managing, this work goals to develop and show the results of a optimization technique (Particle Swarm Optimization - PSO) with two new function of avaliation applied to automatic textual databases clustering and classification problem.

Key-Words: particle swarm optimization, PSO, information retrieval, classification, clustering, base de dados textual

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

PSO – Particle Swarm Optimization

RNA – Rede Neural Artificial

Gbest – Global Best

Lbest – Local Best

Weka – Waikato Environment for Knowledge Analysis

GNU – General Public License

1. INTRODUÇÃO

1.1. Motivação

Cada vez mais as bases de dados geradas por sistemas de informação crescem, dada a geração e armazenamento para análise e manuseio futuro (XU; WUNSCH, 2005). Os dados contidos nessas bases têm suas particularidades podendo ser numéricos, textuais, representação de imagens, de áudio, entre outros.

Os dados textuais representam grande parcela dessas bases de dados, principalmente quando considerados sistemas de informação jornalísticos e para internet.

Em vários desses sistemas de informação, com tanto volume de dados, se torna inviável a extração de informações de forma manual. Para poupar o ser humano de trabalhos árduos foram desenvolvidas: técnicas de limpeza de dados que diminuem o espaço onde trabalhos de extração seriam desenvolvidos, e técnicas de extração de informações propriamente ditas.

As técnicas de limpeza de dados provêm da área de *Data Mining*, que aborda o esforço de trabalhar com grandes volumes de dados para se obter conhecimento valioso e em alto nível (SOUZA et al., 2003). No tratamento de bases textuais algumas das técnicas mais importantes são a eliminação de palavras muito freqüentes ou que não têm significado contextual, conhecidas como *stop-words* (preposições e conjunções, como exemplos), e a redução de expressões para a sua forma normal que é o processo conhecido como *Stemming*.

No processo de conversão da base textual também é intrínseca a transformação desta numa forma computacionalmente tratável. Tal transformação pode ser obtida utilizando o conceito de entropia, descrito e utilizado por Cui (2005) e Wang et al. (2007).

A aquisição de conhecimento através das bases textuais se concentra em encontrar padrões nos dados, ou seja, agrupar as informações e classificá-las de acordo com similaridade e diferenças.

Para que o agrupamento e classificação de dados sejam feitos de forma automática, são utilizadas técnicas de inteligência computacional como é o caso de Algoritmos Genéticos (AG), Redes Neurais Artificiais (RNA) e Partcile Swam Optimization (PSO).

Baseado na revisão bibliográfica realizada foi considerada de grande importância a tentativa de melhora nos resultados obtidos no agrupamento e classificação de dados textuais.

Apresentaremos então a utilização do algoritmo PSO com duas diferentes funções de avaliação propostas por Esmin et al. (2008) e avaliaremos os resultados.

1.2. Objetivos

Objetivamos com o presente trabalho o estudo da técnica PSO, visando à documentação, à comparação com outras técnicas conhecidas e à aplicabilidade desta no agrupamento e classificação de bases textuais. A técnica será qualificada de acordo com sua eficácia na tarefa de classificação utilizando duas novas funções de avaliação introduzidas por Esmin et al. (2008).

1.3. Estrutura do Trabalho

Os capítulos subseqüentes desta monografia estão assim organizados.

No Capítulo 2, são explanados todos os conceitos tomados como necessários para o melhor entendimento do presente trabalho. Dividido em três seções principais, são descritos:

- o problema de tratamento e conversão de bases textuais para uma forma computacionalmente tratável, considerando algumas técnicas conhecidas que se propuseram a resolver o problema de agrupamento e classificação de bases textuais;
- os problemas de agrupamento e classificação de informação, mostrando conceitos básicos da área de conhecimento;
- a técnica de Otimização por Enxame de Partículas (PSO) sendo utilizada para resolução dos problemas de agrupamento e classificação em bases textuais.

No Capítulo 3 descrevemos a metodologia adotada para a elaboração desse trabalho e o processo utilizado para a obtenção de resultados.

No Capítulo 4 são descritos os resultados obtidos de forma analítica em relação à eficácia das técnicas e à aplicabilidade das duas novas funções de avaliação utilizadas no algoritmo PSO.

2. REFERENCIAL TEÓRICO

2.1. Bases de dados textuais

2.1.1. Considerações iniciais

Com o aumento da utilização de sistemas de informação o tamanho das bases de dados textuais cresceu. As informações em forma de texto representam grande fatia dessas bases pelo fato de ser a principal forma de armazenar dados para internet de forma que o ser humano compreenda.

De acordo com Tan (1999), a extração de informações a partir de tais bases de dados é uma tarefa complexa que envolve limpeza dos dados, extração de informação, agrupamento e classificação.

Na etapa de limpeza dos dados, são encontradas técnicas utilizadas para reduzir a complexidade das bases de dados através da normalização de termos (*Stemming*), remoção de termos sem significado contextual (*stopwords*); após a limpeza ainda é necessário convertê-los para uma forma computacionalmente tratável onde temos as operações de cálculo de entropia. (WANG et al., 2007 e CUI et al., 2005).

Nos dias de hoje, com a evolução da pesquisa na área de Inteligência Artificial, foram desenvolvidas técnicas conhecidas como técnicas de otimização que podem buscar por resultados aproximados de problemas que não tenham solução ótima, como em muitos problemas de agrupamento e classificação de informações.

Nesta seção serão descritas algumas técnicas já utilizadas encontradas na literatura que serão utilizadas para comparações de eficácia e eficiência da técnica PSO aplicada ao problema de agrupamento e classificação em bases textuais.

2.1.2. Técnicas de agrupamento e classificação

Muitas técnicas de otimização foram utilizadas para resolver os problemas de agrupamento e classificação de bases textuais como a Rede Neural Artificial (RNA), árvores de decisão e o Particle Swarm Optimization (PSO). Cada uma dessas aborda o problema de agrupamento e classificação de forma distinta.

As Redes Neurais Artificiais foram idealizadas baseadas no funcionamento do cérebro humano (BARRETO, 2002), onde cada neurônio artificial dessa rede dispõe de entradas de dados e sinapses que fazem as ligações com outros neurônios artificiais ou simplesmente com a resposta (saída) do sistema.

De acordo com Barreto (2002), a estrutura de um neurônio pode ser vista na Figura 2.1, onde X_n representam as entradas, W_n os pesos atribuídos respectivamente a cada entrada, θ é uma constante que ajustará o resultado da função Φ, Φ é a função que combinará todas as entradas e pesos do neurônio, η é a função de ativação que produzirá a saída *y* do neurônio.

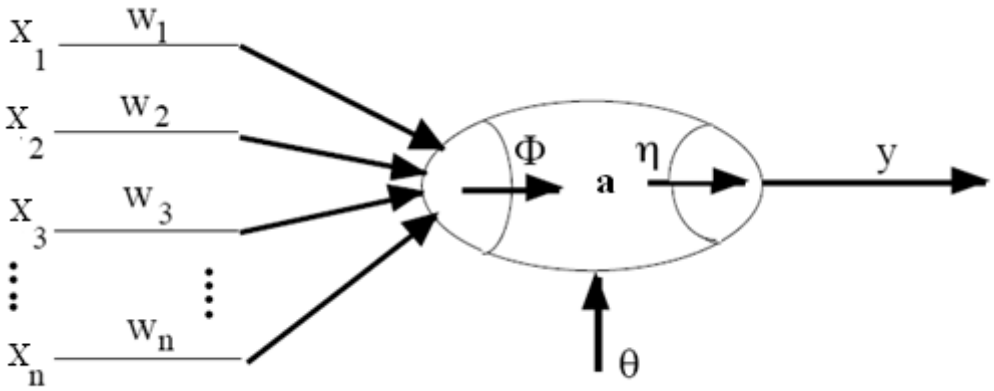

Figura 2.1 – Estrutura básica de um neurônio [Fonte: Barreto, 2002]

O processo de adaptação da rede para melhor representação dos grupos de informação se dá através da atualização dos pesos de cada sinapse (W_n) dando maior ou menor relevância a uma entrada.

Árvores de decisão são outra opção para agrupamento e classificação de dados. De acordo com Quinlan (1992) elas são formadas por uma estrutura simples onde os nós que têm filhos representam testes com um ou mais atributos das amostras submetidas ao algoritmo, e os nós terminais (ditos nós folha) são a representação da decisão a ser tomada, no caso a classe a que uma amostra pertence.

O algoritmo C4.5, proposto por Quinlan (1992), é um algoritmo de árvore de decisão, e foi desenvolvido, dentre outros softwares, dentro do *Weka* (*Waikato Environment for Knowledge Analysis*)(WITTEN e FRANK, 2005) com o nome de J48.

A técnica PSO será descrita mais detalhadamente na Seção 2.3, e juntamente com a RNA e J48 serão aplicadas no presente trabalho visando ao agrupamento e classificação em bases de dados textuais.

2.1.3. Limpeza dos dados

Para fazer uma análise efetiva do conteúdo de documentos, muitos termos se mostram irrelevantes por não apresentarem significado semântico quando considerado separadamente fora de um contexto, como é o caso de interjeições e conjunções da língua portuguesa.

Antes de qualquer atuação de algoritmos de agrupamento e classificação é costume aplicar-se um pré-processamento para a remoção desses termos sem significado semântico, aumentando a eficiência dos algoritmos que trabalharão, então, somente com termos que possam ser efetivamente utilizados nas tarefas de agrupamento e classificação. Wang descreve tais termos que podem ser descartados sem perda de significado a documentos; estes são denominados *Stopwords*.

As *stopwords* são muito utilizadas em ferramentas de buscas disponíveis na internet. Um bom exemplo científico da utilização de *stopwords* pode ser encontrado no Projeto Perseus (MAHONEY, 2000).

De acordo com Mahoney (2000), o Projeto Perseus é uma biblioteca digital evolutiva dos recursos para o estudo das ciências humanas. Colaboradores formavam o projeto para construir um grande e heterogêneo banco de materiais, textuais e visuais, sobre o mundo grego arcaico e Música Clássica. Tal iniciativa teve início em 1985 e foi formalizada em julho de 1987. Desde então, o Projeto Perseus publicou dois CD-ROMs e criou a biblioteca on-line Perseus Digital Library. O projeto tem preparado o caminho para coleções literárias e históricas que vão desde o Renascimento Inglês à Guerra Civil Americana, e uma ferramenta de grego que se tornou uma fundação para o desenvolvimento de recursos em latim, italiano e árabe. Este projeto é mantido pela Tufts University.

A lista de *stopwords* utilizada no Projeto Perseus se encontra no Apêndice A.

2.1.4. Stemming

Almejando ainda a redução da complexidade de problemas que utilizam bases de dados textuais existe a possibilidade de reduzir palavras para suas formas básicas através da remoção de sufixos, processo denominado *Stemming* em inglês.

Porter (1980) descreve a metodologia utilizada na construção de um algoritmo para remoção de sufixos da língua inglesa e exemplifica o processo com remoção dos sufixos de palavras que tem como base a palavra *connect*, como pode ser observado na Figura 2.2.

```
CONNECT
CONNECTED
CONNECTING
CONNECTION
CONNECTIONS
```

Figura 2.2 – Exemplo de Stemming: remoção dos sufixos ED, ING, ION e IONS

Em seus experimentos Porter (1980) conseguiu uma redução do vocabulário de sua base de dados em torno de aproximadamente 33%.

2.1.5. Entropia

De acordo com Wang et al. (2007 apud Guerrero-bote *et. al.*, 2002) uma forma de representação de palavras de maneira computacionalmente tratável é a utilização do conceito de entropia, que associa a um termo em um documento um valor decimal indicando sua relevância na base de dados como um todo. Dumais (1991) expõe o conceito de ponderação por entropia como um método sofisticado de ponderação que utiliza a distribuição de termos em toda a base analisada.

Para a obtenção do valor de entropia de cada termo é necessário inicialmente a construção de uma matriz (Doc_j *x* TF_{jk}) de freqüência dos termos (TF_{jk}) dos documentos (Doc_j). Nesta representação Doc_j se refere a cada documento da coleção de dados e TF_{jk} se refere ao número de ocorrências do termo t_k no documento Doc_j.

Após a construção da matriz de freqüência de termos Cui et al. (2005) utiliza a fórmula descrita na Figura 2.3 para a construção da matriz de entropia dos termos,

$$w_{ij} = tf_{ij} \text{ x } \log (n / df_{ji})$$

Figura 2.3 – Fórmula de cálculo de entropia

onde w_{ij} representa o valor de entropia atribuído a um termo (palavra), tf_{ij} é o número de ocorrências do termo no documento analisado, n é o número total de documentos na base de dados e df_{ji} é a freqüência do termo em toda a base de dados.

2.1.6. Considerações finais

Atualmente muitas técnicas são utilizadas e consideradas satisfatórias para problemas de classificação como exemplo de mensagens indesejáveis (SPAMs). Apesar disso existe a motivação de aplicar as técnicas de agrupamento e classificação automática em problemas mais complexos que possam trazer ganho em termos de eficiência e eficácia proporcionando maior comodidade ao ser humano.

Utilizaremos as técnicas descritas nesse capítulo para a formulação de uma metodologia de agrupamento e classificação de bases de dados textuais.

2.2. Agrupamento e Classificação

2.2.1. Considerações Iniciais

Agrupamento (do inglês, *Clustering*) consiste em formar conjuntos de elementos dentro de um espaço de análise de maneira que os elementos de uma mesma classe sejam o mais semelhante possível.

Agrupamento pode ser definido também como um procedimento exploratório que busca uma estrutura "natural" dentro de um conjunto de dados (COLE, 1998). Os dados são arranjados com base na similaridade entre eles, sem nenhuma suposição sobre a estrutura dos dados.

A disciplina de agrupamento se mostra bastante importante na atualidade principalmente na área de pesquisa de *Data Mining*. Com o agrupamento de dados é possível que sejam extraídas características de similaridade entre os membros de uma classe até então desconhecidas, e ainda a classificação de novas amostras que possam ser submetidas ao modelo de agrupamento gerado.

As aplicações do agrupamento vão desde a área empresarial, como descrito por Han e Kamber (2001) na definição de grupos de clientes baseado em perfis de compra, à área

da medicina para a definição de padrões no surgimento de depressão como descrito por Cole (1998).

Han e Kamber (2001) definem que os métodos de agrupamento podem ser classificados nas categorias Métodos de Particionamento, Métodos Hierárquicos, Métodos Baseados em Densidade, Métodos Baseados em Grid e Métodos Baseados em Modelos.

De acordo com Cui et. al. (2005) as classes mais importantes são a de Métodos de Particionamento e Hierárquicos, por estas duas classificarem a maioria dos algoritmos de agrupamento existentes.

A classe de Métodos de Particionamento é caracterizada pela presença de um conjunto de dados de n elementos e k classes, onde $k \leq n$. O algoritmo é responsável por gerar ou escolher k posições no espaço de solução que se tornarão os centros das classes e distribuir os elementos restantes nas classes através de medidas de similaridade. Nas iterações seguintes é efetuada a realocação de elementos nas partições a fim de aumentar a similaridade entre os elementos de cada classe. Para diminuir a complexidade de tal operação, evitando que o algoritmo percorra todas as possibilidades possíveis (que seria inviável para grandes conjuntos de dados), são utilizadas heurísticas na construção desses algoritmos.

De acordo com Ester et al. (1998), algoritmos da classe de Métodos Hierárquicos representam os conjuntos de dados por um *dendrograma*, uma árvore que iterativamente divide a base de dados em subconjuntos menores até que cada subconjunto consista de somente um objeto.

Na representação das classes, através do método de particionamento, o centro de cada classe é denominado centróide e tem a mesma dimensão de uma amostra analisada pelo problema, ou seja, para análise de dados com três variáveis o centróide de cada classe também tem três dimensões. Ao final do processo de agrupamento dos dados os centróides são a descrição do modelo, sendo utilizados mais a frente para a tarefa de classificação.

2.2.2. Medidas de similaridade

Os valores de similaridades devem medir a distância do documento a ser agrupado ao ponto correspondente ao centróide de cada classe. De acordo com Cole (1998), pequenos valores indicam grande similaridade; então a medida de similaridade pode ser usada para medir dissimilaridade.

Em seu trabalho Cole (1998) descreve três medidas de similaridade, sendo estas, a distância Euclidiana, distância *Manhattan* (i.e. *City Block*) e distância *Minkowski*.

$$d(X_i, X_j) = \sqrt{\sum_{t=1}^{p} (X_{it} - X_{jt})^2}$$

Figura 2.4 – Fórmula da distância Euclidiana

$$d(X_i, X_j) = \sum_{t=1}^{p} |X_{it} - X_{jt}|$$

Figura 2.5 – Fórmula da distância *Manhattan* (*City Block*)

$$d(X_i, X_j) = (\sum_{t=1}^{p} (X_{it} - X_{jt})^n)^{1/n}$$

Figura 2.6 – Fórmula da distância *Minkowski*

As medidas de similaridade acima expostas podem ser aplicadas utilizando as coordenadas do novo documento (pesos da matriz) no espaço de solução e as coordenadas do centróide de cada categoria modelada como sendo os parâmetros X_i e X_j das fórmulas.

2.2.3. Definição Formal de Agrupamento

O problema de agrupamento pode ser definido, de acordo com Cole (1998), da seguinte maneira:

O conjunto de n objetos $X = \{X_1, X_2, X_3, \ldots, X_n\}$ deve ser agrupado. Cada $X_i \in R^p$ é um vetor de p medidas reais que descrevem o objeto. Esses objetos serão agrupados em classes disjuntas $C = \{C_1, C_2, C_3, \ldots, C_k\}$ (C é conhecido como agrupamento), onde k é número de classes, $C_1 \cup C_2 \cup \ldots C_k = X$, $C_i \neq \emptyset$ e $C_i \cap C_j = \emptyset$ para $i \neq j$. Os objetos dentro de cada classe deveriam ser mais similares entre si do que com objetos de outros grupos, e o valor de k pode ser desconhecido. Se k é conhecido o problema é chamado de problema de k-agrupamento.

2.2.4. Agrupamento em bases textuais

O problema central no agrupamento de bases textuais é a modelagem do problema analisado; de forma a tornar a conversão da base textual em um modelo computacionalmente tratável em tempo hábil o mais coerente possível.

Como saída do processo de conversão utilizando limpeza de dados (Seção 2.1.3), *Stemming* (Seção 2.1.4) e cálculo de Entropia (Seção 2.1.5) obtemos uma matriz nas dimensões k x t, onde k é o número de documentos da base de conversão, t é o número total de termos que foram considerados relevantes no pré-processamento da base de dados, e o valor de cada posição dessa matriz é o valor de entropia do termo t no documento k.

A partir dessa conversão podemos aplicar os algoritmos de agrupamento propriamente ditos e ao final do processo de agrupamento, o resultado é um modelo aproximado da melhor representação possível para a base de dados.

2.2.5. Classificação

O processo de classificação consiste em, dado um modelo que representa a forma como os elementos de um conjunto são divididos, atribuir tal elemento ao subconjunto a que este mais se assemelha.

Segundo Manning el al. (2007), para efetuar a classificação é necessária a análise prévia do conjunto de análise (base de conhecimento) para que seja definido o modelo de classificação. Tal análise é feita na etapa de agrupamento onde é definido o modelo que descreve as classes, como descrito na Seção 2.2.3.

Com um modelo que descreve as classes e um elemento que precisa ser classificado, utilizamos as medidas de similaridade (descritas na Seção 2.2.3) onde serão consideradas a posição do elemento e o centróide de cada classe disponível. O elemento será atribuído à classe que tem o centróide a uma menor distância de si.

2.2.6. Classificação em bases textuais

Cada elemento que se deseje classificar segundo o modelo definido na etapa de agrupamento deverá também passar pela conversão do texto para a forma de entropia através das etapas de limpeza de dados (Seção 2.1.3), *Stemming* (Seção 2.1.4) e cálculo de Entropia (Seção 2.1.5).

Já de posse da representação do elemento, é necessário que este seja colocado no mesmo formato dos elementos da base de dados, ou seja, deverá conter o mesmo número de atributos e deverá haver correspondência entre os elementos da base de dados já existente de forma que possamos efetuar comparações entre elementos e centróides com as medidas de similaridade.

2.2.7. Considerações finais

A partir da conversão de um problema com variáveis discretas em variáveis contínuas foi facilitada a aplicação das técnicas de medidas de similaridade já difundidas na literatura que são utilizadas para classificar documentos.

Com a representação em matriz de todos nossos resultados da conversão da base textual da Seção 2.1.3, é possível melhor representar a existência de objetos distribuídos em um espaço R^n que facilita a compreensão do problema de classificação como sendo um problema de cálculo de similaridade através da distância entre objetos.

É possível perceber que o processo de agrupamento é fundamental não só por agrupar os dados manipulados como também serve de base para a criação de um modelo a ser aplicado na classificação de novos documentos que venham fazer parte de nossa base de dados.

Com isso é de muita importância criar um método de agrupamento eficaz para que este não influencie negativamente todos os processos de extração de conhecimento e classificação que possam vir a partir dele.

2.3. Particle Swarm Optimization

2.3.1. Considerações Iniciais

Para obter melhores soluções de problemas de grande complexidade ou inexatos, como é o caso do agrupamento e classificação em bases de dados textuais, podem ser aplicadas técnicas de resolução não lineares, as quais fornecem modelos aproximados do sistema a ser analisado. Nesse contexto são aplicados os algoritmos de otimização, como exemplo a técnica de Otimização por Enxame de Partícula (*Particle Swarm Optimization - PSO*).

O PSO inicia o processo com candidatos a solução do problema (gerado aleatoriamente). Baseando-se na teoria de conhecimento coletivo e influência entre indivíduos aplica-se um critério de qualidade (*fitness*) visando a aproximar o modelo inicial de cada candidato a um modelo que represente melhor o sistema em análise.

Proposto por Kennedy e Eberhart (1995), o PSO se baseia na evolução natural que por sua vez é estudada pela área da Computação Evolutiva.

A otimização é o princípio fundamental por trás da evolução, com o objetivo de sobrevivência das espécies, porém as técnicas de computação evolutiva também podem ser aplicadas em Planejamento, Projeto, Simulação e Identificação, Controle, Classificação (PEREIRA, 2008 apud BERGH, 2001).

2.3.2. Estrutura e Algoritmo

Na concepção original do algoritmo, Kennedy e Eberhart (1995) descreve que existirá uma população a ser gerada com certo número de indivíduos (também conhecidos como partículas) previamente definidos. Cada indivíduo representará uma solução para o problema proposto contendo todos os dados necessários para que possa aproximar a melhor solução.

Para conseguir essa representação são utilizados os conceitos de posição (no espaço de solução), e de velocidade que será responsável por promover a fuga de determinadas posições das partículas consideradas insatisfatórias.

Cada indivíduo da população contém um vetor de velocidades e um vetor de posições onde serão armazenarão os valores atingidos em cada iteração do algoritmo. A velocidade de uma dada partícula será aqui representada por v_j, e a posição por x_j, onde j representa a posição no vetor a qual pertence tal valor.

Durante a execução do algoritmo exposto na Figura 2.9, a posição (ver Figura 2.8) e a velocidade (ver Figura 2.7) serão atualizadas.

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)[y_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}(t)[\tilde{y}_j(t) - x_{i,j}(t)]$$

Figura 2.7 – Fórmula de cálculo de velocidade

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Figura 2.8 – Fórmula de cálculo de posição

Na função de velocidade *c1* e *c2* são valores definidos pelo usuário, *r1* e *r2* são valores gerados aleatoriamente no intervalo de 0 a 1, *y* é a melhor posição local da partícula, *ỹ* é a melhor posição global da execução e *w* representa a inércia do sistema sendo calculado através da função descrita na Figura 2.9 abaixo, na qual são definidos também pelo usuário os valores máximo e mínimo (w_{max} e w_{min} respectivamente).

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \cdot iter$$

Figura 2.9 – Fórmula de cálculo de *w* na função de velocidade

```
Inicializar população

Inicializar partículas (velocidade e posição)

Inicializar o Global Best (da população) e Local Best (de cada
partícula) como o pior possível

For t = 1 to número de iterações Do
Begin

   Avaliar Fitness de cada partícula

   Atualizar o Local Best de cada partícula

   Atualizar o Global Best

   // atualizar velocidade e posição de cada partícula
   For i = 1 to número de partículas Do
   Begin

       Aplicar fórmula de cálculo de velocidade (Figura 2.7)

       Aplicar fórmula de cálculo de posição (Figura 2.8)

   End

End
```

Figura 2.10 – Pseudocódigo do algoritmo PSO

Durante a execução do algoritmo, dados um problema e uma função de avaliação dos resultados (*fitness*), o algoritmo apresentará uma convergência para um bom resultado através do *Gbest*.

Ao final da execução o algoritmo mostrará a melhor partícula (aquela indicada pelo *Gbest*), ou seja, a melhor solução para o problema.

2.3.3. PSO Aplicado ao agrupamento e classificação

Para que possa ser feita a aplicação do PSO à classificação de bases de dados textuais são utilizados os conceitos de tratamento de bases textuais das Seções 2.1.3, 2.1.4 e 2.1.5 para tornar os dados tratáveis computacionalmente em tempo hábil.

Após a limpeza dos dados é aplicada a conversão dos dados através do cálculo de entropia dos termos descrita por Wang et al. (2007, apud Guerrero-bote *et. al.*, 2002) e então montada uma matriz de pesos que será usada para o cálculo de centróides, e posteriormente agrupamento e classificação dessas informações.

Colocando em prática a abordagem de Merwe e Engelbrecht (2003) no modelo PSO, temos que cada partícula representa uma solução completa para o problema. Cada partícula x_i é representada como:

$$x_i = (m_{i1}, m_{i2}, \ldots , m_{ij} , \ldots , m_{iNc})$$

Onde N_c é o número de clusters, m_{ij} corresponde ao j-ésimo centróide da i-ésima partícula.

Para a avaliação das soluções, Merwe e Engelbrecht (2003) propõem também uma função de avaliação (no inglês, *fitness*) que pode ser conferida na Figura 2.11. Onde Z_p denota a p-ésima amostra, $|C_{ij}|$ é o número de amostras pertencentes ao cluster C_{ij} e d é a distância euclidiana entre Z_p e m_{ij}.

$$J_e = \frac{\sum_{j=1}^{Nc} [\sum_{\forall Z_p \in C_{ij}} d(z_p, m_{ij}) / | C_{ij} |]}{Nc}$$

Figura 2.11 – Fórmula de cálculo *fitness* F1 de uma partícula por Merwe e Engelbrecht (2003)

O algoritmo de agrupamento utilizado por Merwe e Engelbrecht (2003) é descrito abaixo na Figura 2.12.

```
Inicializar centróides dos clusters de cada partícula
aleatoriamente

For t = 1 to número de iterações Do
Begin

   For i = 1 to número de partículas Do
   Begin

        For p = 1 to número de amostras Do
        Begin

            Calcular d(Zp, mij) para cada centróide mij

            Atribuir Zp ao cluster Cij tal que:
                d(z_p, m_ij) = min ∀k = 1,...,Nc {d(zp, mik)}

        End

        Calcular o fitness utilizando a função F1

   End

   Atualizar o melhor global e os melhores locais

   Aplicar fórmula de cálculo de velocidade (Figura 2.1)

   Aplicar fórmula de cálculo de posição (Figura 2.2)

End
```

Figura 2.12 – Pseudocódigo do algoritmo PSO aplicado ao agrupamento

Analisando a função F1 para cálculo de *fitness* de Merwe e Engelbrecht (2003), é possível identificar que, no cálculo da média dos valores de distância das amostras a cada centróide, clusters com poucos elementos têm a mesma influência que clusters com muitos elementos.

Esse problema pode ser contornado com equação F2 na Figura 2.13, proposta por Esmin et al. (2008), onde o número de amostras em cada cluster é levado em conta no cálculo da qualidade; nesta função N_0 é o número de amostras a serem agrupadas. A distância média das amostras ao centróide de cada cluster é multiplicada pela porcentagem de amostras que aquele cluster possui. Pereira (2007).

$$F = \{\sum_{j=1}^{Nc}[(\sum_{\forall Z_p \in C_{ij}} d(z_p.m_{ij}) / |C_{ij}|) \times (|C_{ij}| / N_o)]\}$$

Figura 2.13 – Fórmula de cálculo *fitness* F2 de uma partícula por Esmin et al. (2008)

A função de cálculo de *fitness* F1 de Merwe e Engelbrecht ainda apresenta um problema de não favorecer a criação de clusters homogêneos, ou seja, clusters homogêneos, com distribuição mais igualitária de número de amostras entre si.

Para avaliar melhor problemas que tenham dados distribuídos homogeneamente Pereira (2007) propõe a função F3 para o cálculo de fitness. A estrutura da função F3 pode ser vista na Figura 2.14 abaixo.

$$F' = F \times (|C_{ik}| - |C_{il}| + 1)$$

$$\text{Tal que, } |C_{ik}| = \max_{\forall j = 1,..,Nc}\{|C_{ij}|\} \text{ e } |C_{il}| = \min x_{\forall j = 1,..,Nc}\{|C_{ij}|\}$$

Figura 2.14 – Fórmula de cálculo *fitness* F3 de uma partícula por Esmin et al. (2008)

2.3.4. Considerações finais

A técnica de Otimização por Enxame de Partículas se mostrou, através de outros trabalhos estudados, eficaz por apresentar bons resultados tanto na otimização quanto nos problemas de agrupamento e classificação.

Através de modificações no cálculo de *fitness* podemos melhorar ainda mais os resultados no agrupamento de dados, visto que cada problema de agrupamento pode ter suas particularidades.

Com as funções propostas por Pereira (2007), existe em aberto o estudo da aplicação de tais funções para outros problemas, que não os já analisados em seus estudos, sendo uma destas aplicações no agrupamento de bases textuais.

3. METODOLOGIA

De acordo com Jung (2004) a pesquisa, aqui realizada, tem sua natureza classificada como pesquisa básica ou fundamental por gerar conhecimentos e aplicá-los objetivando a solução de um dado problema, e a abordagem ao problema classificada como quantitativa por abordar um problema específico e quantificar os resultados utilizando técnicas estatísticas.

O trabalho foi iniciado com atividades de pesquisa e levantamento bibliográfico, visando à selecionar trabalhos na área de recuperação de informação e otimização, principalmente os que utilizem a técnica PSO na classificação de informação. Tais trabalhos serviram de base para orientar os estudos e desenvolvimento deste trabalho.

Após o levantamento de modelos já utilizados na bibliografia, foi desenvolvido um modelo utilizando a técnica PSO, porém utilizando duas novas funções de avaliação de rendimento, introduzidas por Esmin et al. (2008), com o intuito de efetuar agrupamento e classificação em bases de dados textuais e avaliar a eficácia do PSO com cada uma das novas funções.

Em seguida aplicamos nosso modelo de classificação no *benchmark* Reuter-21578, Distribution 1.0, que é uma coleção de documentos coletados pela *Reuters newswire* em 1987. Possui 5 categorias, 135 subcategorias e 21578 documentos classificados. Então efetuamos comparações entre os resultados, mostrando o quão viável e aplicável é a técnica PSO no contexto proposto.

A Figura 3.1 representa todo esse processo de forma mais detalhada utilizando todos os conceitos vistos no Capítulo 2.

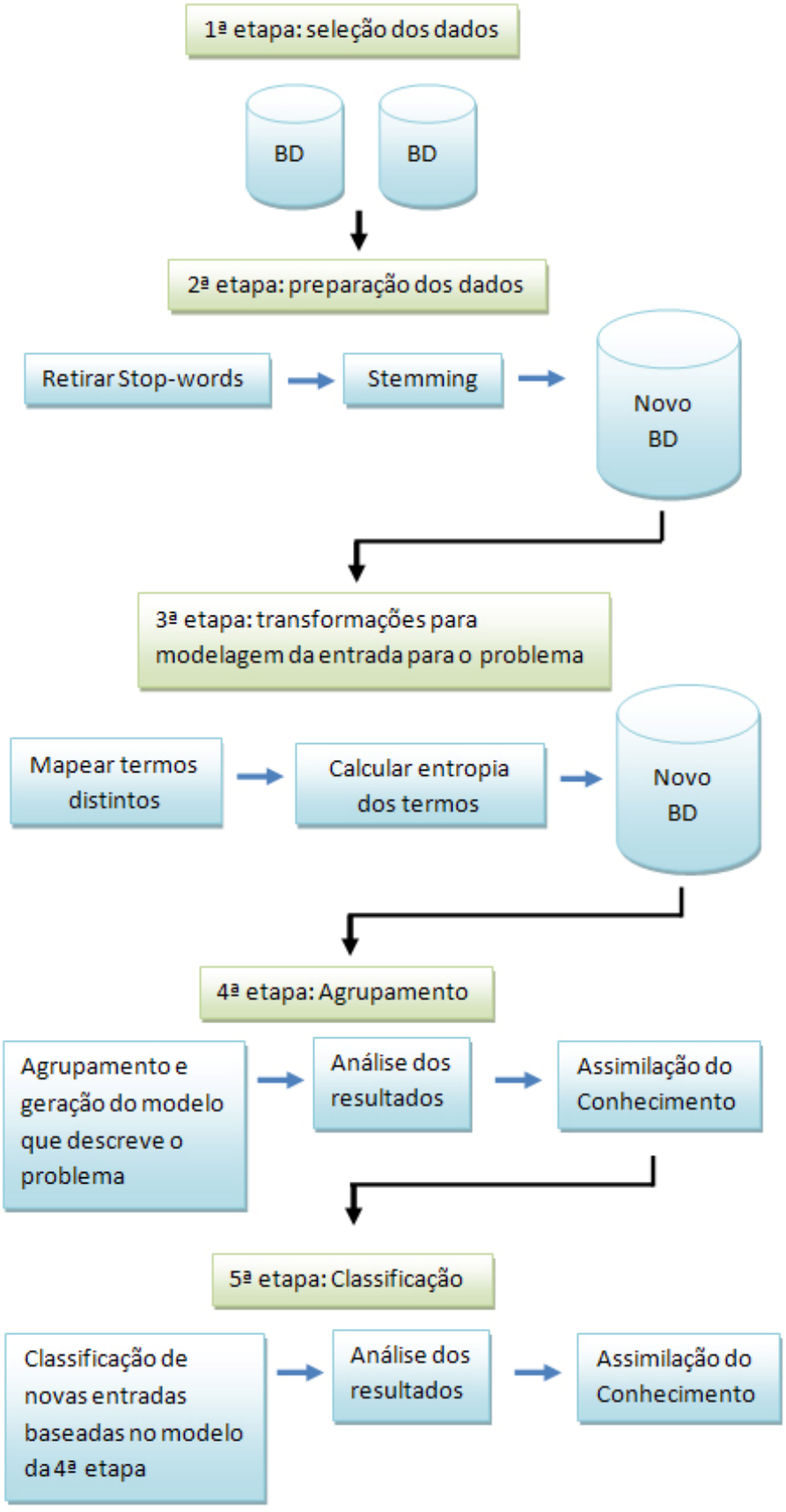

Figura 3.1 – Processo de tratamento de bases de dados textuais

O processo de agrupamento e classificação se inicia com a seleção dos dados (primeira etapa) contidos dentro do banco de dados, ou seja, somente os atributos que se mostrem úteis para o dado objetivo. No caso da base de dados Reuters-21578 foi utilizado somente o campo referente ao corpo principal de cada notícia.

Na segunda etapa, os dados selecionados passaram por um primeiro tratamento: retirar *stopwords* e normalizar expressões (*stemming*).

Para efetuar a remoção de *stopwords* na primeira etapa utilizamos a lista de *stopwords* do Projeto Perseus (MAHONEY, 2000) que segue também no Anexo A do presente trabalho.

Na normalização das expressões utilizamos o algoritmo de Porter (1980) para efetuar o *Stemming*, descrito na Seção 2.1.4. Tal processo reduziu o número de termos a serem considerados no processo de agrupamento e classificação.

A terceira etapa do processo aqui definido é responsável por transformar a base de dados textual em uma forma computacionalmente tratável em tempo hábil. Cada documento passará a ser representado como um vetor de *n* posições, onde *n* é o número de termos distintos na base de dados, e o valor desta posição é a entropia calculada para o dado termo no documento em análise. Para fazer tal representação dos documentos é necessário mapearmos todos os termos distintos na base de dados após os tratamentos efetuados na segunda etapa. Para o cálculo da entropia utilizamos a fórmula proposta por Cui *et al.* (2005), descrita na Seção 2.1.5.

Na quarta e quinta etapas são aplicados os algoritmos de otimização para efetuar o agrupamento e classificação, respectivamente. Em cada processo é possível assimilar conhecimento em relação aos dados analisados, com o principal objetivo de identificar as similaridades entre as amostras agrupadas e classificadas.

4. RESULTADOS E DISCUSSÕES

Para a obtenção de resultados foi necessária a definição de uma metodologia para a realização dos experimentos e o desenvolvimento do modelo PSO em forma de algoritmo pra efetuar os experimentos.

Com o intuito de aumentar a aplicabilidade do estudo em questão desenvolvemos o algoritmo do PSO integrado ao *Weka* (*Waikato Environment for Knowledge Analysis*) (WITTEN e FRANK, 2005). O *Weka* é um software distribuído sob a licença *General Public License* (*GNU*) que contem um grande número de algoritmos para *Data Mining* implementados e interface para diferentes tipos de bases de dados. Dentre os algoritmos implementados no *Weka* estão algoritmos para agrupamento, classificação, seleção de atributos, e diversas formas de visualização dos dados e dos resultados após a execução dos ditos algoritmos. A interface principal da versão para desenvolvedores pode ser conferida na Figura 4.1.

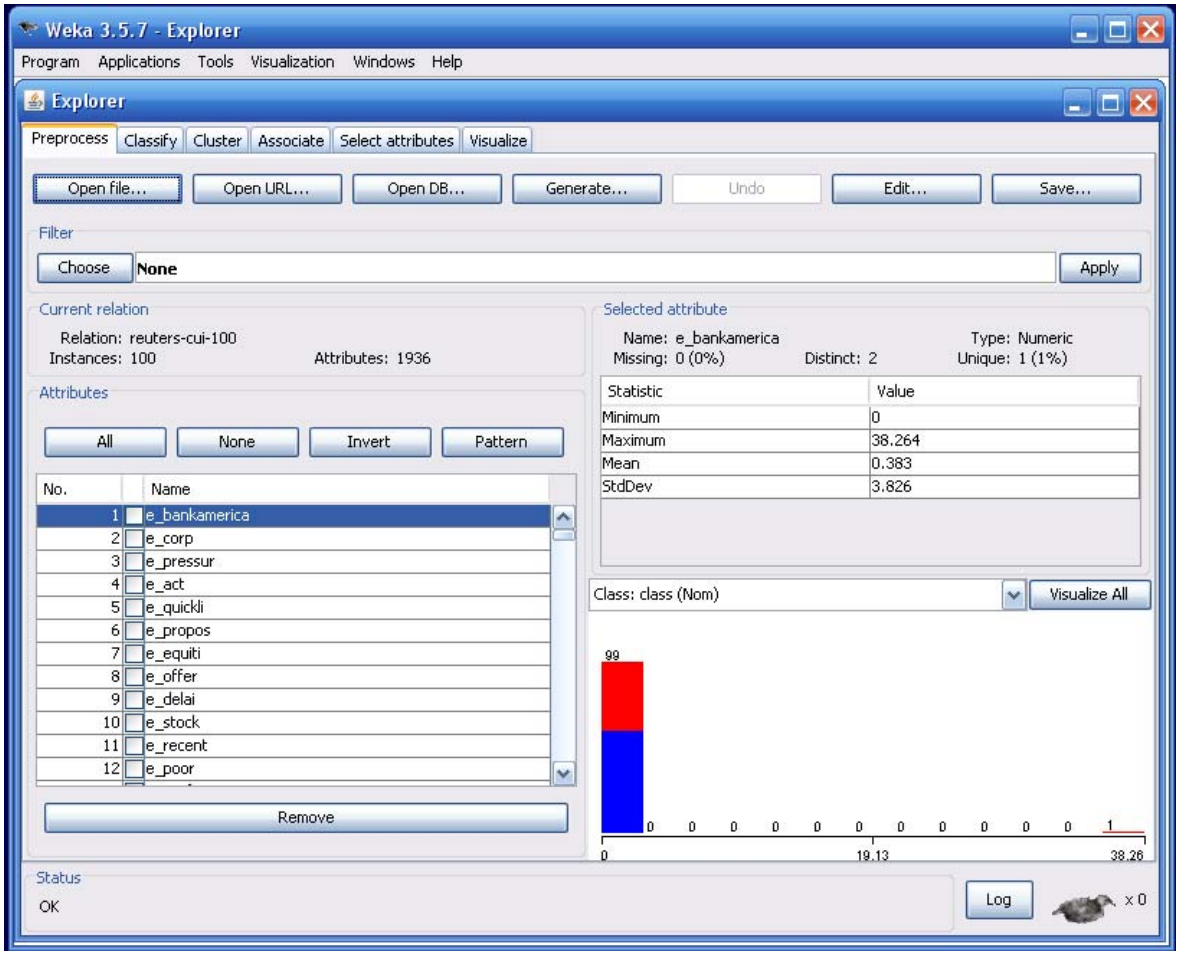

Figura 4.1 – Interface principal da versão de desenvolvimento do *Weka*

Com isso nosso algoritmo foi desenvolvido em Java, de forma orientada à objetos, o que facilitará em muito a manutenibilidade e aprimoramento futuro. Podemos ainda através da interface do PSO, definir parâmetros como o número de partículas, número de clusters e número de iterações. Tal interface pode ser visualizada na Figura 4.2.

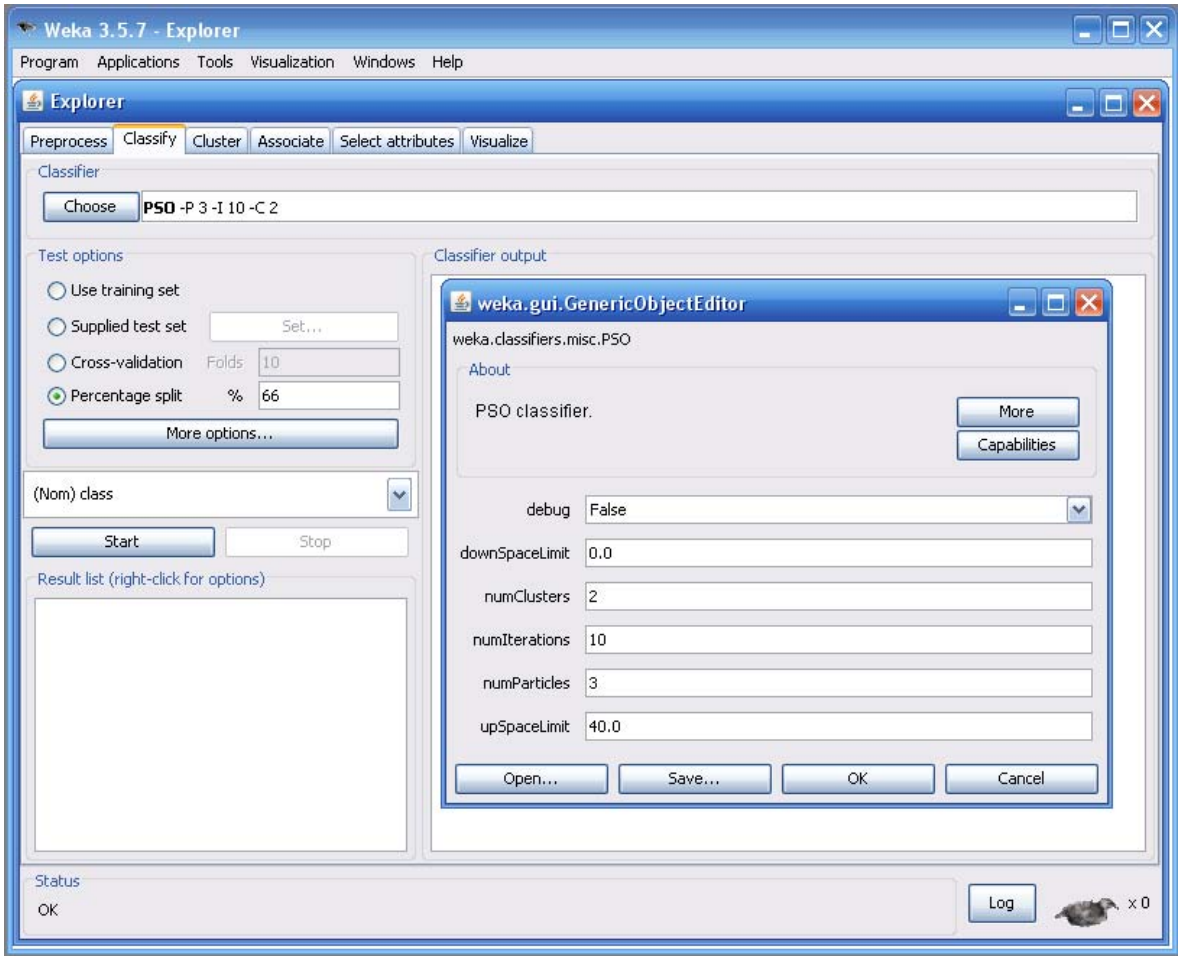

Figura 4.2 – Interface do PSO no software *Weka*

4.1. Resultados

Para avaliar e comparar os resultados utilizamos o *benchmark Reuters-21578*, disponível no *UCI Repository of Machine Learning Databases*. A Tabela 4.1 traz um resumo da base de dados em questão.

Tabela 4.1 – Atributos da base de dados *Reuters-21578*

Categorias (Classes)	5
Subcategorias	135
Número de documentos	21578

Para a obtenção dos resultados e comparações foram utilizados os algoritmos PSO, Rede Neural Artificial de Base Radial (denominada também como *RBFNetwork*) e J48, ambos descritos na Seção 2.1.2. As tabelas 4.2, 4.3 e 4.4 mostram os parâmetros utilizados por cada algoritmo em sua execução.

Tabela 4.2 – Parâmetros de execução do algoritmo PSO

Número de partículas	3, 5, e 10
Número de clusters	2
Número de iterações	40
Divisão da base de dados (Treinamento/Validação)	66% / 34%
w (na função velocidade)	De 1 a 0
c1 (na função velocidade)	1.49
c2 (na função velocidade)	1.49

Tabela 4.3 – Parâmetros de execução do algoritmo Rede Neural Artificial (RBFNetwork)

Número de clusters	2
Divisão da base de dados (Treinamento/Validação)	66% / 34%
Erro máximo	0.1

Tabela 4.4 – Parâmetros de execução do algoritmo J48

Divisão da base de dados (Treinamento/Validação)	66% / 34%
Mínimo de instancias por folha	2

A base de dados utilizada para os experimentos foi selecionada aleatoriamente utilizando distribuição uniforme na escolha das amostras da base de dados completa. Selecionamos então 100 amostras, que durante o processo de limpeza e transformação da base de dados gerou amostras com 1936 atributos.

Cada experimento, com os parâmetros descritos nas Tabelas 4.2, 4.3 e 4.4, foi executado dez vezes e selecionado a melhor execução de acordo com a porcentagem de amostras classificada corretamente na etapa de validação. No caso do algoritmo PSO, foram executados tais experimentos com as funções F1, F2 e F3 (descritas respectivamente nas Figuras 2.11, 2.13 e 2.14) e população de 3, 5 e 10 partículas. Por restrições recursos computacionais não foi possível executar experimentos com o PSO com população maior que 10 partículas. A Tabela 4.5 descreve os resultados obtidos.

Tabela 4.5 – Resultados da melhor execução de cada algoritmo

Atributo do resultado	**PSO F1**	**PSO F2**	**PSO F3**	**RNA**	**J48**
Amostras classificadas corretamente	22 (64,705 %)	22 (64,705 %)	22 (64,705 %)	24 (70,58 %)	18 (52,94 %)
Amostras classificadas incorretamente	12 (35,294 %)	12 (35,294 %)	12 (35,294 %)	10 (29,41 %)	16 (47,05 %)
Erro Médio Absoluto	0,1412	0,1412	0,1412	0,1176	0,1902
Erro Quadrático relativo	121,9435 %	121,9435 %	121,9435 %	111,31 %	132,23 %

A variação de população não afetou a análise através dos melhores resultados, sendo que todos os experimentos acertaram 22 amostras (máximo descrito na tabela) pelo menos uma vez.

Calculamos ainda a média de todas as dez execuções, das quais foi possível obter os resultados da Tabela 4.6.

Atributo do resultado	**PSO F1 Pop=3**	**PSO F1 Pop=5**	**PSO F1 Pop=10**	**PSO F2 Pop=3**	**PSO F2 Pop=5**	**PSO F2 Pop=10**	**PSO F3 Pop=3**	**PSO F3 Pop=5**	**PSO F3 Pop=10**	**RNA**	**J48**
Amostras Classif. Correta_ mente	17,7 (52,058 %)	15,1 (44,411 %)	15,6 (45,882 %)	17 (50 %)	18 (52,941 %)	17 (50 %)	17 (50 %)	14 (41,176 %)	19 (55,882 %)	24 (70,58 %)	18 (52,94 %)
Amostras Classif. Incorreta_ mente	16,3 (47,941 %)	18,9 (55,588 %)	18,4 (52,941 %)	17 (50 %)	16 (47,058 %)	17 (50 %)	17 (50 %)	20 (58,823 %)	15 (44,117 %)	10 (29,41 %)	16 (47,05 %)
Erro Médio Absoluto	0,1917	0,2223	0,21646	0,2	0,18824	0,2	0,2	0,23528	0,17648	0,1176	0,1902
Erro Quadrático relativo (%)	140,65	151,89	149,73	143,52	139,21	143,52	143,52	156,47	134,89	111,31	132,23

Tabela 4.6 – Resultados médios de execuções dos algoritmos

Nas figuras abaixo expomos a evolução média do melhor *fitness* em toda a população de cada experimento utilizando o PSO com a função *fitness* F1, F2 e F3. Os dados apresentam intervalo de *fitness* diferentes dada a formula de cálculo de *fitness*. É possível visualizar que o PSO com as funções F2 e F3 de avaliação apresentam maior velocidade de convergência para os respectivos melhores resultados.

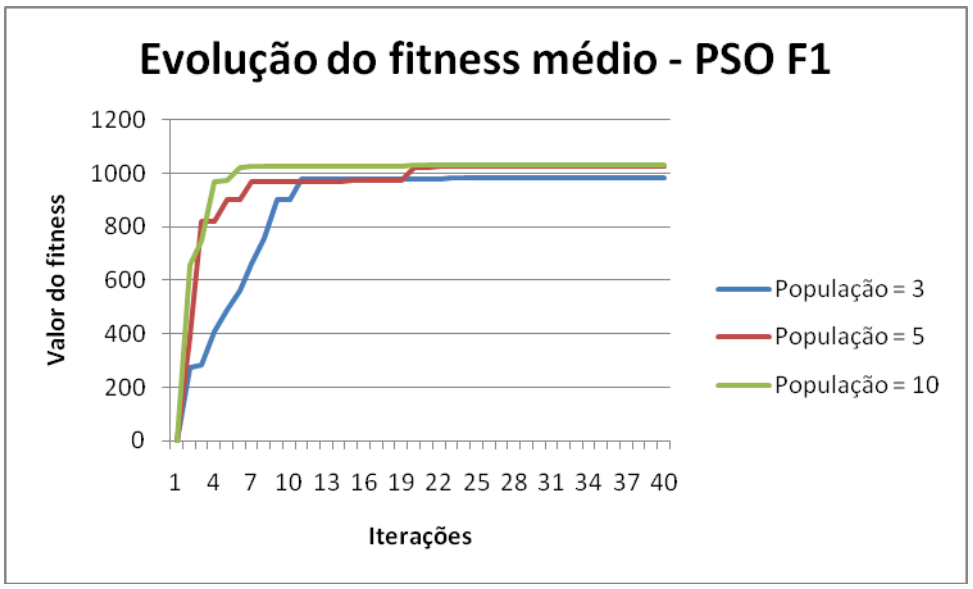

Figura 4.3 – Evolução do *fitness* médio para PSO F1 com variação de população

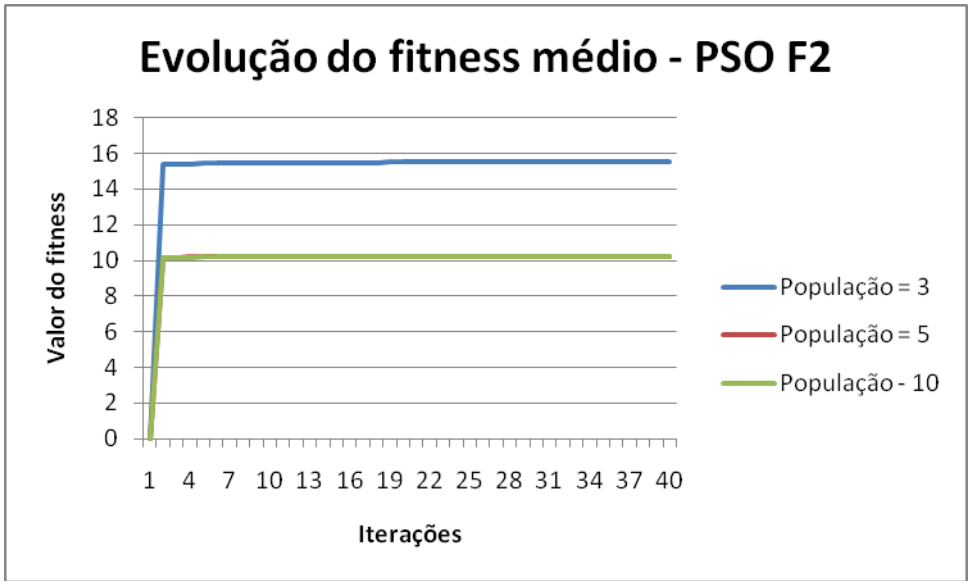

Figura 4.4 – Evolução do *fitness* médio para PSO F2 com variação de população

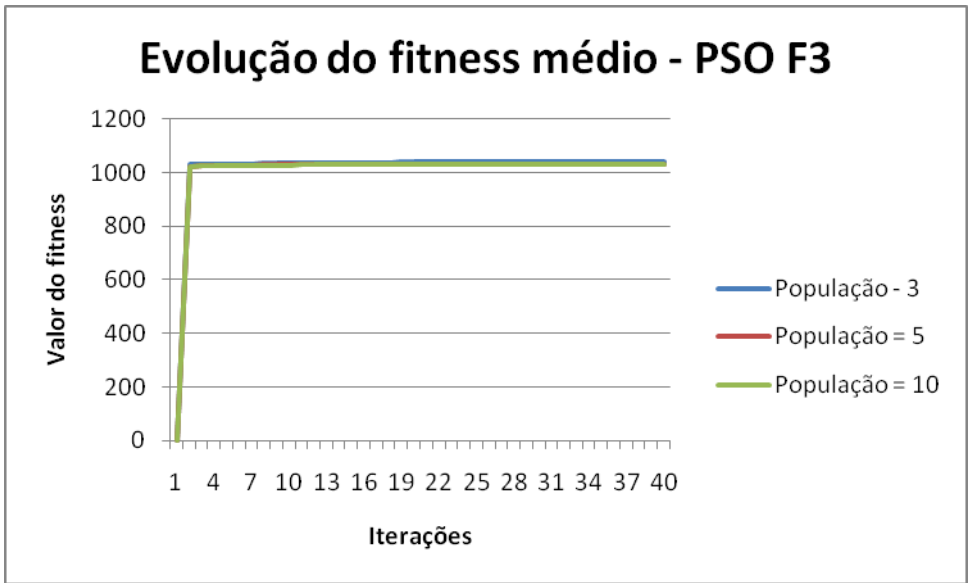

Figura 4.5 – Evolução do *fitness* médio para PSO F3 com variação de população

Como pode ser observado, a Rede Neural Artificial obteve melhores resultados, tanto quanto à melhor execução quanto a média de todas execuções, apresentando as menores taxas de erro na classificação.

O algoritmo J48 obteve baixos índices de acerto em sua melhor execução, porém obteve melhores resultados médios em comparação com todos os experimentos do PSO, com exceção do PSO F3 com população de 10 partículas.

Dentre os algoritmos PSO, todas as funções para cálculo de *fitness* obtiveram resultados iguais em suas melhores execuções, distante da Rede Neural artificial em somente uma amostra classificada errada, ou seja, 11 classificadas erradas pela RNA e 12 pelo PSO.

Os resultados médios mostram que o PSO com função de cálculo de *fitness* F3 com população de 10 partículas obteve maior índice de acertos, 1,3% superior ao PSO com função de cálculo de *fitness* F1 em sua melhor configuração de população (3 partículas).

4.2. Conclusão

Neste trabalho foi possível desenvolver o algoritmo PSO no software livre de *Data Mining* chamado *Weka* (*Waikato Environment for Knowledge Analysis*) utilizando a linguagem de programação Java e Orientação a Objetos.

Definimos uma metodologia, baseada na literatura, para ser seguida no processo de agrupamento e classificação em bases de dados textuais, utilizando conceitos de limpeza de dados, pré-processamento e transformação de bases de dados.

Por fim aplicamos a metodologia na base de dados *Reuters-21578*, e executamos o algoritmo PSO com três diferentes funções de avaliação de rendimento e variação de população (3, 5 e 10 partículas), e outros dois algoritmos conhecimentos (Rede Neural Artificial de Base Radial e J48).

Com os resultados da execução dos algoritmos foi possível perceber que o algoritmo Rede Neural Artificial de Base Radial (*RBFNetwork*) teve melhor rendimento nas tarefas de agrupamento e classificação, e que a diferença para o PSO com todas as funções de cálculo de *fitness* se mostra significativa quando considerados os resultados médios. Isso nos dá como motivação o aprofundamento dos estudos no PSO com tais funções de avaliação para que possamos adaptar os parâmetros do algoritmo visando a uma melhora nos resultados, visto que o PSO é uma técnica relativamente nova e bem flexível dada a possibilidade de modificação dos parâmetros para problemas específicos.

4.3. Contribuições

O presente trabalho contribuiu com conteúdo experimental, produto de software e conteúdo teórico.

Como conteúdo experimental, expomos os resultados do PSO com parâmetros específicos para o problema de agrupamento e classificação de bases de dados textuais.

Como produto de software, desenvolvemos o algoritmo PSO dentro do software livre *Weka*, o que possibilitará o desenvolvimento, manutenção e utilização contínua do PSO em *Data Mining* pela comunidade internacional.

Sobre conteúdo teórico disponibilizamos a metodologia empregada com o intuito de promover a continuidade dos estudos, principalmente no que diz respeito ao amadurecimento de uma metodologia que torne o tratamento de bases textuais um processo menos caro computacionalmente.

4.4. Trabalhos Futuros

A partir deste estudo identificou-se a necessidade de realizar um estudo sobre a adequação automática dos parâmetros da função velocidade do PSO.

Percebe-se também a necessidade de aprimorar a metodologia empregada no agrupamento e classificação em bases textuais. Dada que a complexidade de tal problema é geralmente grande, é interessante comparar resultados com a aplicação de técnicas de seleção de atributos antes de efetuar o agrupamento das amostras.

REFERENCIAL BIBLIOGRÁFICO

ARAUJO, E.; COELHO, L. S.. **Piecewise fuzzy reference-driven Takagi-Sugeno modelling based on particle swarm optimization**. Capítulo em Systems Engineering using Swarm Particle Optimisation. Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches (Chapter 12). N. Nedjah; L. Mourelle, New York, NY. Nova Science Publishers, 2006.

BARRETO, J. B.. **Introdução às Redes Neurais Artificiais**. UFSC, 2002.

BERGH, F. van der. **An Analysis of Particle Swarm Optimizers**. 2001. 300 p. Tese (PhD in the Faculty of Natural and Agricultural Science) – University of Pretoria, Pretoria.

COHEN, W. W.. **Fast Effective Rule Induction**. Twelfth International Conference on Machine Learning, 1995.

COHEN, W. W. **Learning rules that classify e-mail**. In Papers from the AAAI Spring Symposium on Machine Learning in Information Access, PP. 18-25. 1996.

COLE, R. M. **Clustering with Genetic Algorithms**. 1998. 110 p. Tese (Master of Science). University of Western Australia , Perth.

CUI, X.; POTOK, T. E.; PALATHINGAL, P.. **Document Clustering using Particle Swarm Optimization**. IEEE Swarm Intelligence Symposium, June, 2005, PP 185- 191, Pasadena, California, Estados Unidos da América.

CUI, X.; POTOK, T. E.. **Document Clustering Analysis based on Hybrid PSO+K-means Algorithm**. The Journal of Computer Science, 2005 volume1(3), pp 27 – 33, Science Publications, New York, Estados Unidos da América.

DOLAMIC, L.; SAVOY, J.. **Stemming Approaches for East European Languages**. Cross-Language Evaluation Forum (CLEF 2007), 2007, Budapeste, Hungria.

DUMAIS, S. T.. **Improving the retrieval of information from external sources**. Behavior Research Methods, Instruments and Computers. IngentaConnect Publication, Cambridge, MA, Fevereiro de 1991, pp. 229-236.

ESMIN, A. A. A. ; PEREIRA, D. L. ; ARAÚJO, F. P. A.. **Study of Different Approach to Clustering Data by Using The Particle Swarm Optimization Algorithm**. In: IEEE Congress on Evolutionary Computation (IEEE CEC 2008), 2008, Hong Kong. Proceedings of IEEE World Congress on Computational Intelligence (WCCI), 2008.

ESMIN, A. A. A. ; TORRES, GERMANO LAMBERT ; ALVARENGA, G. B.. **Hybrid Evolutionary Algorithm Based on PSO and GA Mutation**. In: 6th International Conference on Hybrid Intelligent Systems (HIS06), 2006, Auckland. IEEE Computer Society, 2006. PP. 57-60.

ESTER, M. ; KRIEGEL, H. P. ; SANDER, J. ; WIMMER, M. ; XU, X**.. Incremental Clustering for Mining in a Data Warehousing Environment**. In: Proceedings of the 24th International Conference on Very Large Data Bases (VLDB), pp. 323-333, New York City, New York, USA, August. 1998.

GUERRERO-BOTE, V. P.; MOYA-ANEGON, F.; HERRERO-SOLANA, V.. **Document organization using Kohonen's algorithm**. Information Processing and Management. Elsevier Sicence, New York, 2002. PP. 79-89.

HAN, J.; KAMBER, M. **Data Mining**: Concepts and Techniques. San Francisco: Morgan Kaufmann, 2001. 550 p.

JUNG, C. F. **Metodologia para Pesquisa & Desenvolvimento.** Rio de Janeiro, RJ. Axcel Books do Brasil Editora, 2004.

KENNEDY, J.; EBERHART, R. C. Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, 1995, Perth. **Proceedings of IEEE International Conference on Neural Networks**. 1995, p. 1942- 1948.

MAHONEY, A.. **Explicit and implicit searching in the Perseus digital library**. In Einat Amitay, editor, Information Doors: Pre-Conference Workshop at the Eleventh ACM Conference on Hypertext and Hypermedia, 2000.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H.. **An Introduction to Information Retrieval**. Cambridge University Press, 2007.

MERWE, D. W. van der; ENGELBRECHT, A. P. Data Clustering using Particle Swarm Optimization . In: Congress on Evolutionary Computation, 2003. **Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003)**, Caribella: IEEE Computer Society, 2003. p. 215-220.

PEREIRA, D. L.. **Estudo do PSO na clusterização de dados**. Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação. 2007.

PORTER M. F.. **An Algorithm for Suffix Stripping.** Vol. 3, No. 14., pp. 130-137. Plain, ACS - American Chemical Society, 1980.

QUINLAN, J.R.. **C4.5 Programs for Machine Learning**. San Mateo, CA: Morgan Kaufmann, 1992.

RUSSEL, S.; NORVIG, P.. **Inteligência Artificial**. Campus, 2003. Tradução da 2ª Edição.

SOUZA, T.; NAVES, A.; SILVA, A. Swarm Optimisation as a new tool for data mining. In: Parallel and Distributed Processing Symposium, 17., 2003. **Proceedings of the 17th International Symposium on Parallel and Distributed Processing**. IEEE Computer Society, 2003.

TAN, A. H.. **Text Mining: The state of the art and the challenges.** In Proceedings of the PAKDD 1999, Workshop on Knowledge Discovery from Advanced Databases, 1999.

WANG, Z.; ZHANG, Q.; ZHANG, D.. **A PSO-Based Web Document Classification Algorithm**. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007, Qingdao, China.

WITTEN, I. H.; FRANK, E.. **Data Mining: Practical machine learning tools and techniques**, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

XU, R.; WUNSCH, D.. **Survey of Clustering Algorithms**. IEEE Transactions on Neural Networks, Vol. 16, No 3. Maio 2005.

APÊNDICE A

Disponível em: http://www.perseus.tufts.edu/Texts/engstop.html

Último acesso em 08/10/2008.

Tabela A.1 – Lista de stopwords

a	about	above
accordingly	after	again
against	ah	all
also	although	always
am	an	and
and/or	any	anymore
anyone	are	as
at	away	b
be	been	begin
beginning	beginnings	begins
begone	begun	being
below	between	but
by	c	ca
can	cannot	come
could	d	did
do	doing	during
each	either	else
end	et	etc
even	ever	far
ff	following	for
fro	from	further
get	go	goes
going	got	had
has	have	he
her	hers	herself
him	himself	his
how	i	if
in	into	is
it	its	itself
last	lastly	less
ll	many	may
me	might	more
must	my	myself
nay	near	nearly
never	new	next
no	not	now
o	of	off
often	oh	on

only	or	other
otherwise	our	ourselves
out	over	over
perhaps	put	puts
quite	s	said
saw	say	see
seen	shall	she
should	since	so
some	such	t
than	that	the
their	them	themselves
then	there	therefore
these	they	this
those	thou	though
throughout	thus	to
too	toward	unless
until	up	upon
us	ve	very
was	we	were
what	whatever	when
where	which	while
who	whom	whomever
whose	why	with
within	without	would
yes	your	yours
yourself	yourselves	