Sistema Computacional de Codificação
Automática de Atividades Econômicas

Projeto Classificação Automática em CNAE-Subclasses

Relato de Cumprimento de Metas No. 3

Meta Física 1.1/2007
Meta Física 1.2/2007
Meta Física 1.3/2007
Meta Física 1.4/2007
Meta Física 1.5/2007
Meta Física 2.1/2007
Meta Física 3.1/2007
Meta Física 3.2/2007
Meta Física 4.1/2007

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Sumário

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Índice de Figuras

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Índice de Tabelas

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

1 Introdução

Este documento descreve as atividades levadas a cabo visando o cumprimento das Metas 1/2007, 2/2007, 3/2007 e 4/2007, previstas em convênio firmado em dezembro de 2006 entre a Receita Federal do Brasil (Receita) e a Fundação Espírito-Santense de Tecnologia (FEST). Este convênio (Convênio Receita/FEST) tem como objeto a cooperação tecnológica entre os partícipes visando ao desenvolvimento de estudos e pesquisas que subsidiem a análise da aplicabilidade de técnicas de Inteligência Computacional na classificação automática de documentos, sendo os documentos de interesse descrições de atividades econômicas dos agentes econômicos e a classificação de interesse a identificação de códigos da tabela CNAE-Subclasses (Classificação Nacional de Atividades Econômicas – Subclasses Fiscais) correspondentes às atividades econômicas descritas nos documentos. Por meio deste convênio foi viabilizada a realização do Projeto de Pesquisa "Classificação Automática em CNAE-Subclasses".

Acompanha este Relato um DVD com os códigos e bases de dados desenvolvidos, além de outros documentos relevantes.

As redundâncias observáveis entre este documento e o "Relato de Cumprimento de Metas No. 1" (SCAEa, 2007) e o "Relato de Cumprimento de Metas No. 2" (SCAEb, 2007), referentes ao mesmo Projeto, visam apenas permitir uma leitura autocontida do presente documento. Nos documentos anteriores podem ser encontrados complementos e detalhamentos de modelos aqui apresentados e desenvolvidos.

1.1 Motivação e justificativa

No XVI Encontro Nacional de Auditores e Fiscais de Tributos Municipais, ocorrido em 8 e 9 de Julho de 2004 em Manaus, constatou-se que os municípios perdem cada vez mais receitas e albergam cada vez mais atribuições. A correta interpretação e aplicação das leis tributárias municipais é fonte inequívoca de mais e melhores receitas para o município. Uma das formas de se melhorar os mecanismos de arrecadação é a adoção de um sistema consistente de codificação tributária.

O inciso XXII do Art. 37, incluído na Constituição por emenda constitucional em dezembro de 2003, versa que "as administrações tributárias da União, dos Estados, do Distrito Federal e dos Municípios (...) atuarão de forma integrada, inclusive com o compartilhamento de cadastros e de informações fiscais, na forma da lei ou convênio". No 1º Encontro Nacional de Administradores Tributários (ENAT), realizado em Salvador – BA, em julho de 2004, e com a participação das secretarias de fazenda dos estados, capitais e da Receita, foi iniciada a discussão sobre as iniciativas necessárias para o cumprimento desse dispositivo através de ações conjuntas. Neste cenário de atuação integrada, a adoção da CNAE-Subclasses assumiu um papel mandatório para a implementação dos cadastros compartilhados.

No modelo atual, a atribuição de códigos a agentes econômicos segundo a CNAE-Subclasses (classificação em CNAE-Subclasses) é realizada manualmente por diversos agentes codificadores (contabilistas, funcionários de órgãos públicos, etc...) e em diversas fases da existência do agente econômico, o que resulta numa baixa qualidade de codificação. Neste

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

contexto, o desenvolvimento de uma ferramenta automática para tal finalidade se apresenta como uma estratégia bastante promissora para atacar este problema.

Assim, a principal motivação para a realização do Projeto de Pesquisa "Classificação Automática em CNAE-Subclasses" é a oportunidade que ele oferece de se investigar a solução de um problema prático específico, mas cujas técnicas de solução têm aplicação em diversos outros cenários relevantes. Outra motivação importante é a possibilidade que ele oferece de estender o estado da arte em ciência da informação (classificação de texto), ciência da computação (classificação automática de texto, computação de alto desempenho) e ciência da cognição (representação de conhecimento). Este trabalho se justifica pela importância econômica e governamental da correta, eficiente e eficaz classificação de atividades econômicas.

1.2 Objetivos

O objetivo principal do Projeto de Pesquisa "Classificação Automática em CNAE-Subclasses" é desenvolver ou adaptar algoritmos e heurísticas que viabilizem a implementação de um protótipo de um Sistema Computacional de Codificação Automática de Atividades Econômicas (SCAE) e comparar o desempenho deste sistema com o de codificadores humanos. Também são objetivos deste Projeto implementar um protótipo de um SCAE com interface Web, expandir o estado da arte nas áreas de pesquisa associadas ao projeto e formar pessoal especializado.

1.3 Organização deste documento

Após esta introdução, na Seção 2 apresentamos brevemente o problema científico de interesse, isto é, a classificação automática em CNAE-Subclasses. Na Seção 3, relatamos o cumprimento das Metas Físicas 1.1/2007, 1.2/2007, 1.3/2007, 1.4/2007, 2.1/2007, 3.1/2007, 3.2/2007 e 4.1/2007 (a Meta 4.2/2007 foi relatada no Relato de Cumprimento de Metas No. 1). Na Seção 4 apresentamos outras realizações técnico-científicas associadas ao Projeto e, na Seção 5, breve relato da participação da equipe científica em eventos relevantes. Por fim, na Seção 6, discutimos importantes lições aprendidas desde a apresentação do Relato de Cumprimento de Metas No. 2.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

2 Classificação Automática em CNAE-Subclasses

A Classificação Nacional de Atividades Econômicas (CNAE), uma tabela hierárquica de atividades econômicas e seus códigos associados, é a classificação oficialmente adotada pelo Sistema Estatístico Nacional e pelos órgãos federais gestores de registros administrativos associados. Com base na Resolução do Presidente do IBGE No. 054, de 19/12/1994, publicada no Diário Oficial da União No. 244, em 26/12/1994, vem sendo implementada desde 1995 pelo Sistema Estatístico Nacional e órgãos da administração federal. A CNAE foi desenvolvida tendo por referência a *International Standard Industrial Classification of All Economic Activities* - ISIC, 3ª revisão, das Nações Unidas. O responsável pela gestão e manutenção da CNAE é o IBGE, a partir das deliberações da Comissão Nacional de Classificação – CONCLA. A partir da elaboração da CNAE, que hoje contempla 672 classes de atividades (CNAE 2.0), foi derivada outra classificação, a CNAE-Subclasses.

A CNAE-Subclasses é um detalhamento das classes da CNAE para uso nos cadastros da administração pública, em especial da administração tributária, nas três esferas do governo. Sua estrutura é igual à estrutura da CNAE, que possui código de 5 dígitos para cada classe, adicionada de um nível hierárquico, codificado com 2 dígitos. Ou seja, as subclasses da CNAE-Subclasses possuem código de 7 dígitos, sendo os dois dígitos adicionais resultantes do detalhamento de cada classe da CNAE. Este detalhamento foi feito especificamente para atender necessidades da organização dos cadastros de pessoas jurídicas no âmbito das três esferas da Administração Pública. Na CNAE 2.0 existem 1301 subclasses.

A CNAE-Subclasses é usada como instrumento de padronização nacional dos códigos de atividade econômica utilizados pelos diversos órgãos públicos da administração direta, facilitando a gerência e o controle de ações da competência de cada esfera. Nos cadastros da administração tributária o código CNAE-Subclasse é atribuído a todos os agentes econômicos que estão engajados na produção de bens e serviços, podendo compreender estabelecimentos de empresas privadas ou públicas, estabelecimentos agrícolas, organismos públicos e privados, instituições sem fins lucrativos, e agentes autônomos (pessoa física). Na Secretaria da Receita Federal, um ou mais códigos CNAE-Subclasse devem ser informados na Ficha Cadastral de Pessoa Jurídica (FCPJ) quando do cadastro de uma nova pessoa jurídica ou quando da alteração dos seus atos constitutivos – a FCPJ alimenta o Cadastro Nacional de Pessoa Jurídica (CNPJ) da Receita Federal do Brasil.

Atualmente, em muitos órgãos usuários, a determinação de quais códigos devem ser atribuídos a cada agente econômico, a codificação em CNAE-Subclasses, é feita manualmente por codificadores humanos treinados para tal, apoiados por ferramentas computacionais de busca em versões eletrônicas da tabela CNAE-Subclasses. O codificador humano treinado deve associar/combinar a informação na tabela CNAE-Subclasses com seu conhecimento, fruto de seus vários anos de educação e experiência profissional, para, com o conjunto, atribuir códigos CNAE-Subclasse para o agente econômico cujas atividades estão sendo codificadas.

Na verdade, para fazer a codificação, o codificador humano precisa compreender quais são as atividades do agente econômico e qual é a correspondência entre elas e os descritores de uma ou mais Subclasses da CNAE-Subclasses. Para operar com eficácia equivalente, um Sistema

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Computacional para a Codificação Automática de Atividades Econômicas (SCAE) precisa gerar representações da tabela CNAE-Subclasses e das atividades do agente econômico, internas ao sistema. Estas representações têm que ser tais que permitam identificar a correta correspondência semântica entre a descrição das atividades do agente econômico e um ou mais subclasses/descritores da tabela CNAE-Subclasses. Mecanismos para o enriquecimento da representação interna da tabela também devem ser incorporados ao SCAE de modo a prover um equivalente computacional dos vários anos de educação e experiência profissional do codificador humano.

A codificação automática em CNAE-Subclasses de interesse compreenderá, então, a identificação do(s) descritor(es) de atividade(s) e respectivo(s) código(s) CNAE-Subclasses de um agente econômico a partir:

- das descrições completas de suas atividades econômicas;
- da Tabela CNAE-Subclasses e seus instrumentos de apoio à codificação;
- de uma base de dados representativa de codificações corretas; e
- de um dicionário eletrônico da língua portuguesa e outras bases de dados que permitam criar uma representação interna ao computador das atividades do agente econômico e da tabela CNAE-Subclasses ricas o suficiente para permitir ao computador identificar as correspondências entre as atividades e as entradas da tabela.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

3 Metas Físicas Alcançadas

No período de dezembro de 2006 a julho de 2007 foram alcançadas as Metas Físicas:

Meta Física 1.1/2007 – Desenvolvimento de mecanismo de extração atividades econômicas no formato interno ao sistema de contratos sociais ou de descrição de atividades econômicas

Meta Física 1.2/2007 – Desenvolvimento de mecanismo de codificação baseado em redes neurais artificiais

Meta Física 1.3/2007 – Desenvolvimento de mecanismo de codificação baseado em redes Bayesianas

Meta Física 1.4/2007 – Desenvolvimento de mecanismo de codificação baseado em *Latent Semantic Indexing*

Meta Física 1.5/2007 – Desenvolvimento de mecanismo de composição dos resultados da codificação através de neurais artificiais, redes Bayesianas e *Latent Semantic Indexing* em uma única codificação, mais robusta

Meta Física 2.1/2007 – Implementação de protótipo do SCAE-Fiscal

Meta Física 3.1/2007 – Criação de *benchmarking* para realização de comparações entre os métodos

Meta Física 3.2/2007 – Avaliação estatística dos mecanismos de codificação desenvolvidos

Meta Física 4.1/2007 – Realização de seminários de acompanhamento e avaliação

Estas metas foram previstas no Plano de Trabalho do Convênio Receita/FEST para o período de junho de 2007 a fevereiro de 2008. É importante observar que: (i) devido ao deslocamento de dois meses do segundo desembolso e de nove meses do terceiro desembolso previstos no Plano de Trabalho, o cronograma do projeto foi também deslocado; e que (ii) a Meta Física 4.2/2007, prevista para o período de agosto de 2007 a abril de 2008, foi relatada no documento “Relato de Cumprimento de Metas No. 1” (SCAEa, 2007).

A seguir entregamos os resultados físicos associados a cada meta (códigos, relatórios, bases de dados, e documentos relativos a seminários realizados), conforme previsto no Plano de Trabalho do Convênio Receita/FEST. Este Relato é acompanhado de um DVD com os códigos e bases de dados desenvolvidos, além de outros documentos relevantes.

3.1 Meta Física 1.1/2007: Desenvolvimento de Mecanismo de Extração Atividades Econômicas no Formato Interno ao Sistema de Contratos Sociais ou de Descrição de Atividades Econômicas – Fundamentação do Código

Tipicamente, descrições de atividades econômicas são apresentadas ao SCAE para classificação segundo a Tabela CNAE na forma de texto livre. Para que o SCAE as classifique, é necessário que elas sejam primeiramente transformadas para um formato próprio para a interpretação por sistemas computacionais – o formato interno ao sistema. Para os mecanismos de classificação implementados, a forma de representação interna apropriada para o SCAE é a forma vetorial.

Para o SCAE, descrições de atividade econômica na forma de texto livre são documentos. Neste relato, para comodidade do leitor, reapresentamos a forma vetorial de representação de documentos. Em seguida, apresentamos a representação de documentos interna ao SCAE e a facilidade de filtro do SCAE, empregada para tratar acentos, realizar correção ortográfica, entre outras finalidades relevantes.

3.1.1 Representação Vetorial de Documentos

No Modelo Vetorial de representação de documentos, os documentos são representados por vetores no espaço R^n (BAEZA-YATES e RIBEIRO-NETO, 1998). Onde n representa o número de palavras distintas nos documentos considerados. Cada documento é considerado, portanto, um vetor de ocorrência de palavras.

Formalizando o que foi dito acima, consideremos um conjunto de documentos $D = \{d_1, d_2, ..., d_i, ...d_n\}$, onde d_i, um dos elementos deste conjunto, é um documento representado por um vetor de pesos $d_i = [w_1, w_2, ..., w_k, w_{k+1}, w_{k+2}, ...w_n]$, sendo que k é o número de palavras $\{t_1, t_2, ..., t_k\}$ distintas que aparecem no documento d_i. As demais palavras $\{t_{k+1}, t_{k+2}, ..., t_n\}$, associadas aos pesos $\{..., w_{k+1}, w_{k+2}, ..., w_n\}$, são palavras que aparecem em outros documentos. Assim, $\{t_1, t_2, ...t_k, t_{k+1}, t_{k+2}, ..., t_n\}$ são todas as palavras do vetor que representa o documento d_i e a freqüência das palavras $t_{k+1}, t_{k+2}, ..., t_n$ é igual a zero. Deste modo, podemos concluir que, no modelo vetorial de representação de documentos, uma palavra pode aparecer em mais de um documento.

Classicamente, o peso w_i de importância de uma palavra está relacionado à ocorrência dessa palavra t_i tanto no próprio documento onde a palavra ocorre, como também o número de ocorrências desta em outros documentos. A seguir apresentamos as formas de representação da importância das palavras dos documentos no SCAE.

3.1.2 Representação de Documentos Interna ao SCAE

Os textos dos documentos não podem ser interpretados diretamente por um algoritmo de classificação. Procedimentos, que mapeiam um texto d_j numa representação compacta, precisam ser aplicados uniformemente aos documentos. O SCAE trata duas propostas (procedimentos) de ponderação da importância de palavras: *tf* (*term frequency*) e a *tfidf* (*term frequency-inverse document frequency*).

Na proposta *tf*, o peso w_i de importância de uma palavra é calculado pela freqüência da palavra t_i no próprio documento d_j, ou seja, contabiliza-se o número de vezes que a palavra t_i ocorre no documento d_j. Nessa proposta, as palavras com freqüência baixa, que possuem relevância para discriminar um documento de outro, recebem pesos menores, e as palavras com freqüência alta, que não são relevantes entre documentos, recebem pesos maiores.

Na proposta *tfidf*, o peso w_i é calculado levando em consideração a freqüência da palavra t_i no próprio documento d_j, proposta *tf*, e a freqüência da palavra t_i nos documentos do conjunto *D*, isto é, o número de documentos em *D* que a palavra t_i ocorre. Essa abordagem é definida pela função mostrada na Equação 3-1.

$$tfidf(t_i, d_i) = \#(t_i, d_i) * \log(|D| / \#D(t_i))$$

Equação 3-1

onde: o termo $\#(t_i,d_i)$ representa a *tf*; o $\#D(t_i)$ representa o número de vezes que a palavra t_i ocorre em *D*, chamada de *df* (*document frequency*); o $log(|D|/\#D(t_i))$ representa a freqüência inversa da palavra em *D*, chamada *idf* (*inverse document frequency*).

A proposta *tfidf* codifica a intuição que (i) quanto mais freqüente uma palavra num documento, maior é a importância semântica dela para o próprio documento, e (ii) quanto mais freqüente uma palavra no conjunto documentos, menor é o poder de discriminação dela.

Tanto a formulação *tf* quanto a formulação *tfdif* levam em consideração apenas a ocorrência das palavras, não considerando a ordem na qual elas aparecem nos documentos e o papel sintático que elas possuem.

3.1.3 Facilidade de Filtro do SCAE

Nem sempre a forma na qual palavras aparecem em documentos é a mais apropriada para classificação automática de textos. Para converter documentos para o formato vetorial o SCAE primeiramente os filtra.

O conhecimento lingüístico pode, principalmente através de processamento morfossintático, trazer estratégias inteligentes para a classificação de texto, como, por exemplo, exploração de técnicas de normalização de variações lingüísticas, eliminação de *stopwords*[1] e a correção ortográfica.

A seguir, um maior detalhamento sobre estas técnicas. O algoritmo radicalizador, que implementa uma forma de normalização de variações lingüísticas empregada, é apresentado na Seção 3.1.3.3, e o processo de sua inclusão no SCAE, mais precisamente no DB_CORE, é descrito na Seção 3.1.3.4. Na Seção 3.1.3.5, a operação de filtro, que implementa a eliminação de *stopwords*, é apresentada, equanto que, na Seção 3.1.3.6, a operação de correção ortográfica é apresentada.

3.1.3.1 Normalização de Variações Lingüísticas

A normalização lingüística pode ser subdividida em três casos distintos (ARAMPATZIS, 2000): morfológica, sintática e léxico-semântica:

- Normalização morfológica: ocorre quando há redução dos itens lexicais através de conflação (melhor definida abaixo) a uma forma que procura representar classes de conceitos;
- Normalização sintática: ocorre quando há a normalização de frases semanticamente equivalentes, mas sintaticamente diferentes, em uma forma única e representativa das mesmas, como fruta madura e saborosa e fruta saborosa e madura.
- Normalização léxico-semântica: ocorre quando são utilizados relacionamentos semânticos (sinonímia, hiponímia e meronímia) entre os itens lexicais para criar um agrupamento de similaridades semânticas, identificado por um item lexical que representa um conceito único.

Cabe mencionar que utilizamos somente a normalização morfológica.

Conflação

A conflação é o ato de fusão ou combinação para igualar variantes morfológicas de palavras. Ela pode ser manual, usando algum tipo de expressão regular, ou automática, via programas chamados radicalizadores (*stemmers*). Para reduzir as variações de uma palavra para uma forma única utilizam-se técnicas de conflação, por exemplo, a radicalização.

[1] De acordo com a literatura pesquisada na língua portuguesa, será usado o termo na língua inglesa e não uma tradução, pois não existe tradução consagrada.

Radicalização

Radicalização (*stemming*) é o processo de combinar as formas diferentes de uma palavra em uma representação comum, o radical (*stem*) (OREGANO, 2001). Radical é o conjunto de caracteres resultante de um processo de radicalização. Este não é necessariamente igual à raiz lingüística, mas permite tratar variações diferentes de uma palavra da mesma forma. Por exemplo, conector e conectores são essencialmente iguais, mas sem sofrerem a redução por radicalização serão tratadas como palavras distintas.

Freqüentemente, é especificada uma palavra em uma consulta, mas somente uma variante desta palavra é apresentada em um documento relevante. Plurais, formas de gerúndio e sufixos de tempos verbais são exemplos de variações sintáticas que impedem uma perfeita combinação entre uma palavra de consulta e uma palavra do respectivo documento. Por exemplo, as palavras durabilidade, duradouro e durável poderiam ser reduzidas para a representação comum dur-.

Este método já é amplamente usado em processamento de textos para Recuperação de Informação baseado na suposição de que uma consulta com o termo durável implica num interesse em documentos que contenham também as palavras durabilidade e duradouro. Este problema é solucionado com a substituição de palavras pelos seus respectivos radicais.

Segundo Porter (PORTER, 1980), radicalização é o processo de remoção das terminações morfológicas e flexionais das palavras.

3.1.3.2 Eliminação de *Stopwords*

Stopwords são palavras freqüentes em um texto e que não representam nenhuma informação de maior relevância para a extração de palavras-chave. Por exemplo: advérbios, artigos, conjunções, preposições e pronomes. As *stopwords* freqüentemente não fornecem nenhuma contribuição na identificação do conteúdo do texto. A remoção das *stopwords* tem como objetivo eliminar palavras que não são representativas do documento e isso, conseqüentemente, diminui o número de palavras a serem analisadas no mesmo, e também o número de palavras a serem armazenadas em uma base de busca de informações.

3.1.3.3 *Portuguese Stemmer*

Dentre os algoritmos radicalizadores disponíveis para a língua portuguesa, optou-se pelo *Portuguese Stemmer*, o qual foi proposto por Viviane Orengo e Christian Huyck (ORENGO, 2001). Este algoritmo leva em conta as classes morfológicas, executando uma série de passos de remoção de sufixos conhecidos. Os passos são aplicados na seguinte seqüência:

1. Redução do plural;
2. Redução do feminino;
3. Redução do advérbio;
4. Redução do aumentativo e diminutivo;
5. Redução das formas nominais;
6. Redução das terminações verbais;

7. Redução da vogal temática;
8. Remoção dos acentos.

A figura abaixo ilustra a seqüência dos passos.

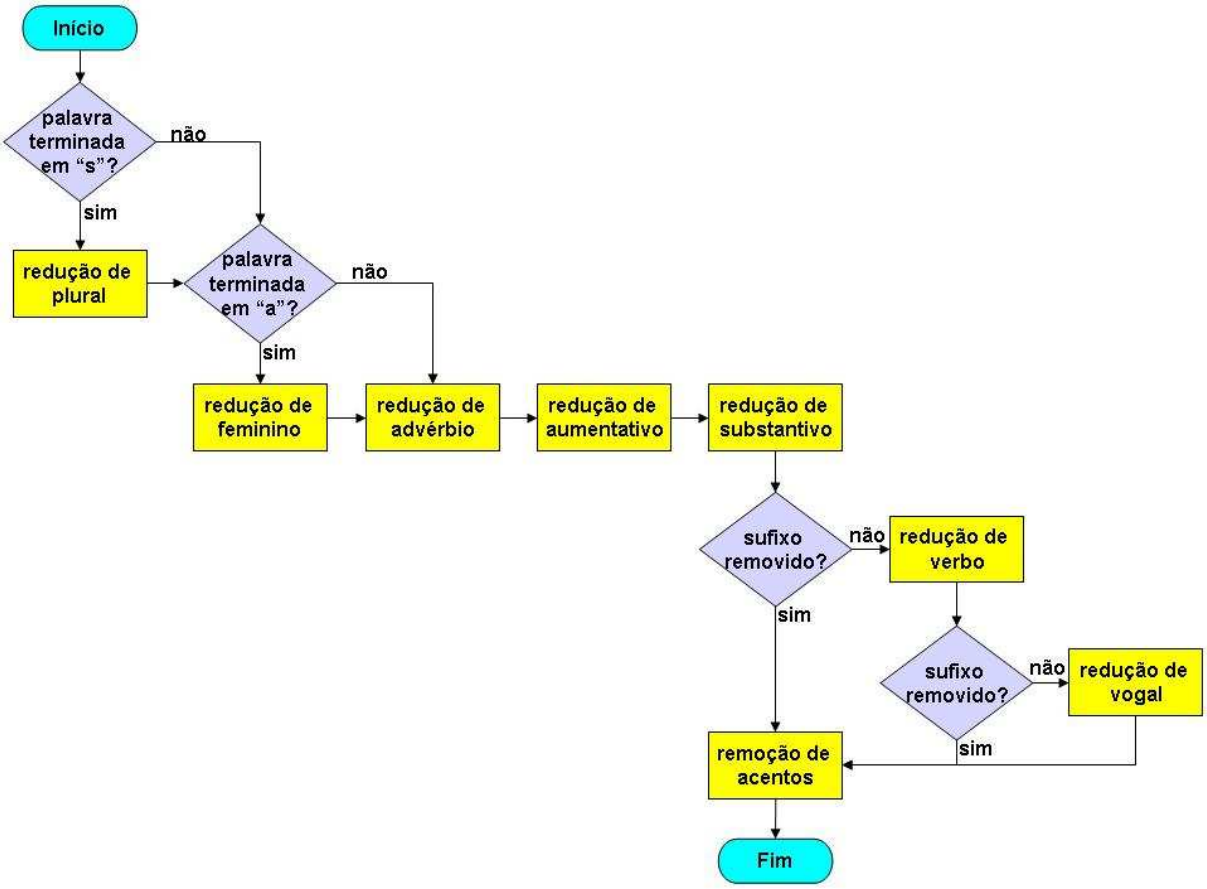

Figura 3-1: Os passos do *Portuguese Stemmer*.

3.1.3.4 Inclusão no SCAE

Felizmente, havia uma implementação do *Portuguese Stemmer*, em Java, disponível em (DIAS 2008). Contudo, não bastava simplesmente embutir o código no SCAE, mais precisamente no DB_CORE, uma vez que o banco de dados foi codificado em C. Neste ponto, existiam dois caminhos possíveis a fim de se obter uma solução para a comunicação entre os códigos:

- Converter o código do radicalizador para C;
- Manter a codificação e utilizar JNI (*Java Native Interface* (LIANG, 1999) – padrão de programação que permite que aplicações Java sejam embutidas em aplicações nativas, no caso, codificadas em C);

Optou-se por manter o código do radicalizador em Java, pois era uma forma de adquirir conhecimento para inclusão de futuros CORES (por exemplo, o *BN – Bayesian Network*, que também está codificado em Java).

A implementação disponível continha, também, uma lista de *stopwords* em Java (contendo 336 termos). Esta lista de *stopwords*, juntamente com o radicalizador, compuseram o filtro *stemmer* do DB_CORE. A figura abaixo apresenta o algoritmo deste filtro.

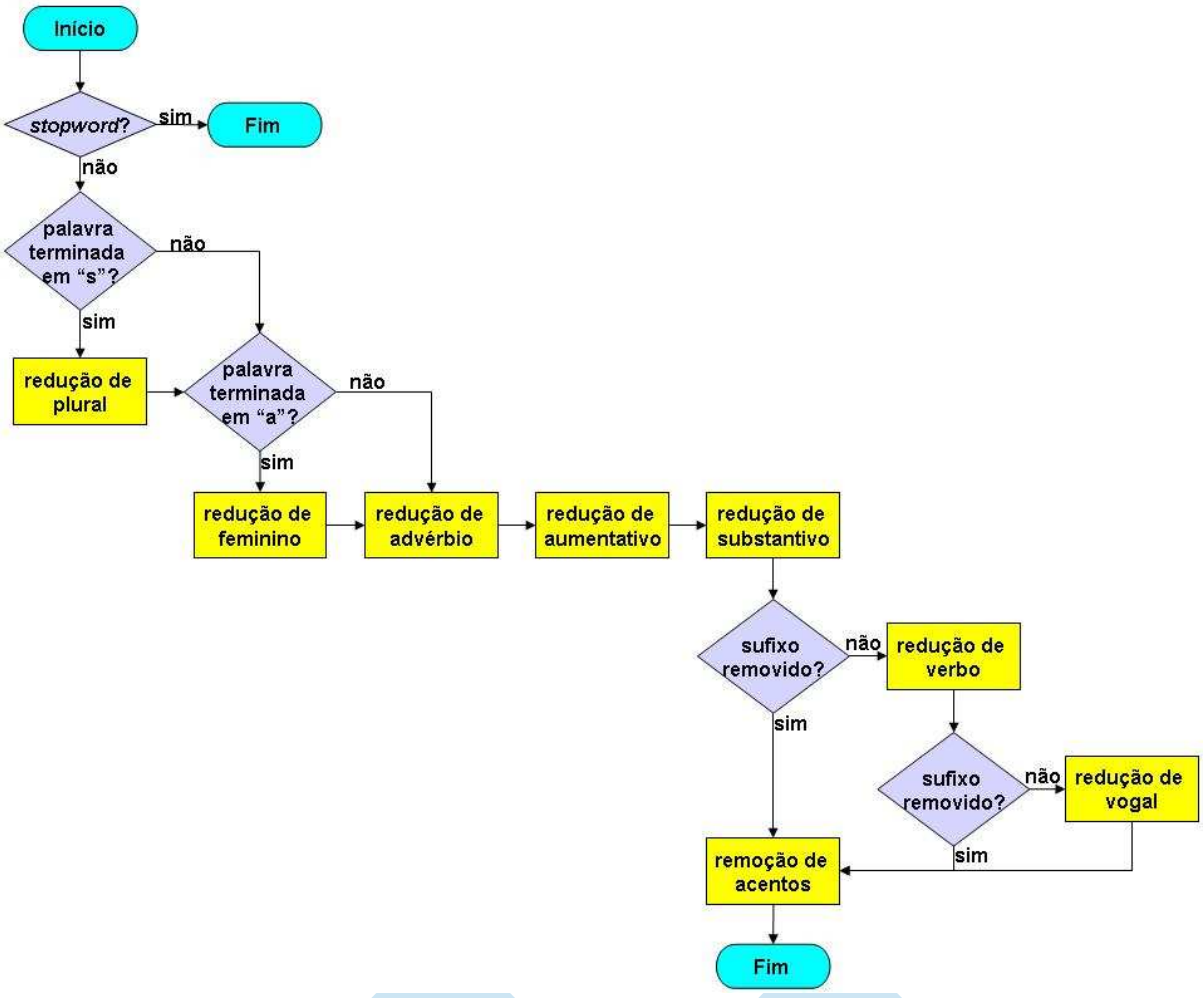

Figura 3-2: Os passos do filtro adicionado ao SCAE.

Observe que a única diferença entre os fluxogramas do *Portuguese Stemmer* (Figura 3-1) e o filtro do SCAE (Figura 3-2) é a existência de uma tomada de decisão ("*stopword*?") no início do filtro adicionado ao SCAE (novamente, ao DB_CORE).

3.1.3.5 Operação do Filtro

À primeira vista, desejar-se-ia que o filtro da Figura 3-2 fosse invocado para cada palavra quando da construção de um vetor de treino e teste (o significado deste vetor, bem como a sua

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

estrutura foram apresentados em Relatos anteriores). Porém, esta abordagem seria muito custosa devido ao *overhead* inserido pela JNI.

Optou-se, então, por aplicar o filtro em um momento anterior ao da criação dos vetores de treino e teste, como é apresentado na Seção Pré-Processamento, a seguir. Na Seçao Impacto sobre o DB_CORE, o impacto do filtro sobre o DB_CORE é relatado.

Pré-Processamento

O pré-processamento consiste simplesmente em criar um dicionário filtrado, o qual possui a mesma estrutura daquele gerado pelo Diadorim (aplicativo fornecido pelo NILC – USP). As diferenças residem em:

- No lugar da palavra canônica existirá a palavra radicalizada;
- Se alguma das opções de filtragem (detalhadas melhor a seguir) for a de remoção de *stopwords*, a palavra não será inserida no dicionário;
- Diferentemente da redução à forma canônica (também apresentada em relatórios anteriores), na radicalização há perda da categoria morfológica original, isto é, um radical pode ser oriundo de palavras de categorias diferentes. Por exemplo, se as palavras salvando e meninas fossem radicalizadas, estas seriam transformadas nos respectivos radicais salv - e menin-, e se as mesmas palavras fossem reduzidas à forma canônica seriam representadas como salvar e menino, respectivamente.

As etapas do pré-processamento, isto é, da criação do dicionário filtrado, são esboçadas no fluxograma abaixo. Novamente, deve ser frisado que a etapa de construção dos vetores de treino e teste não é executada neste momento.

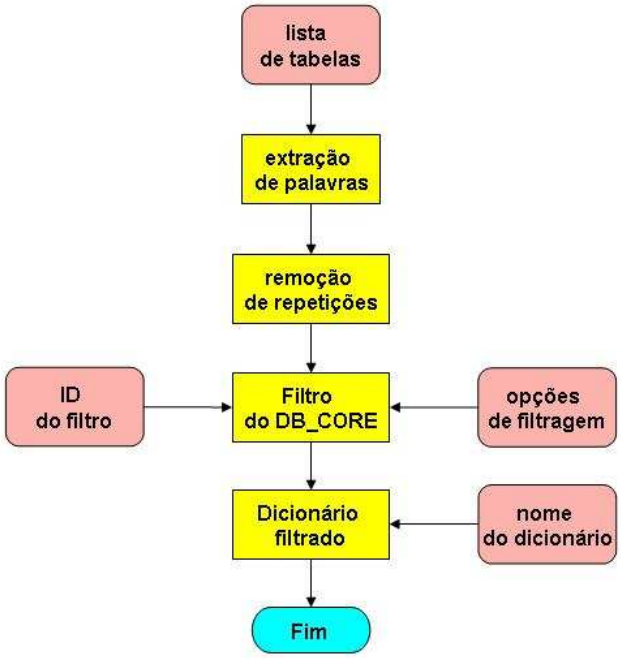

Figura 3-3: Os passos para criação de um dicionário filtrado.

Até o momento existe um único filtro (apresentado na Figura 3-2, cujo ID é igual a 1). Novos filtros podem ser facilmente incorporados sem exercer impacto sobre o DB_CORE, devido à modularidade do código.

Impacto sobre o DB_CORE

O impacto da inclusão da funcionalidade de filtro sobre o DB_CORE foi nulo. O processo de criação de vetores de treino e teste, bem como as funções preexistentes do DB_CORE permaneceram inalterados.

Ao usar um dicionário pré-existente (por exemplo, um gerado pelo aplicativo fornecido pelo NILC – USP chamado Diadorim), cada elemento de um vetor gerado pelo processo de construção de tabelas representará o peso de uma palavra canônica. No caso do emprego de um dicionário filtrado (pseudo-dicionário, como vimos anteriormente), cada elemento, deste mesmo vetor, representará o peso de uma palavra filtrada (radicalizada).

Conclui-se, então, que a opção por construir vetores a partir de palavras filtradas recai na seleção de um dicionário filtrado apropriado, o qual, obrigatoriamente, foi construído em uma etapa prévia.

3.1.3.6 Corretor Ortográfico

As máquinas de busca (*Search Engines*) tornaram-se o principal meio de acesso às informações na *Web*. Entretanto, recentes estudos mostram que palavras escritas de forma errada nas consultas para esses sistemas são muito comuns. Dalianis, em (DALIANIS, 2002), mostra que o percentual de consultas com palavras escritas incorretamente está entre 10 e 12%.

Os erros de escrita também ocorrem nas descrições das atividades econômicas de empresas, pois as mesmas são textos livres descritos pelos seres-humanos. Tais erros podem afetar o desempenho de categorizadores automáticas como o SCAE. Então, existe a necessidade de se realizar um pré-processamento nos documentos das bases de dados utilizados como treino e teste com a finalidade de detectar e corrigir tais erros. Esse pré-processamento é realizado no SCAE pelo Corretor Ortográfico.

Como as bases de dados de interesse possuem um grande número de documentos, a correção manual torna-se custosa. Então, o corretor ortográfico foi implementado de forma a selecionar automaticamente a primeira (melhor escolha) entre todas as possíveis correções para a palavra errada, em vez de mostrar uma lista de possíveis correções para o humano escolher qual a palavra correta. Para fazer a melhor escolha, o dicionário faz uso de um dicionário e uma lista de palavras com as respectivas freqüências.

Técnicas de Correção Ortográfica

Conforme Martins (MARTINS, 2004), a correção ortográfica está relacionada a dois principais problemas: detecção de erro, que é o processo de encontrar a palavra errada; e a correção de erro, que é o processo de sugerir palavras corretas para a errada.

Martins classifica erros ortográficos em duas categorias: erros de tipografia e erros fonéticos. Os erros de tipografia ocorrem por causa de uma digitação errada ao pressionar uma tecla errada, pressionar duas teclas, inverter a ordem das letras, etc; e erros fonéticos, onde palavra

é escrita da mesma forma que a sua pronúncia, mas com erros de ortografia. Os erros fonéticos são difíceis de corrigir porque eles distorcem a palavra com mais de uma única inserção, supressão ou substituição.

Hoje, os corretores ortográficos são ferramentas comuns para diversos idiomas, e muitas propostas podem ser encontradas na literatura. A maioria dessas ferramentas é baseada em dicionário. Os métodos propostos nessas ferramentas incluem *edit distance*, *rule-based techniques*, *n-grams*, *probabilistic techiniques*, *neural nets*, e *similarity key techiniques* (MARTINS, 2004). Basicamente, todas essas técnicas consistem em calcular a distância entre a palavra errada e cada uma das palavra no dicionário. A de menor distância é colocada no topo da lista de sugestões das possíveis palavras corretas.

Conforme Martins, a maioria das palavras erradas nos textos, cerca de 80 a 95%, difere das palavras corretas em um dos quatro modos mais comum de erro de ortografia, que são: *transposition* (transposição), *wrong letter* (letra errada), *extra letter* (letra extra) e *missing letter* (omissão de letra). Esses modos são apresentados na Figura 3-4. As palavras superiores são as do dicionário, representadas pela linha *Dictionary*, e as palavras inferiores são digitadas pelo usuário com erro de ortografia, representadas pela linha *User*.

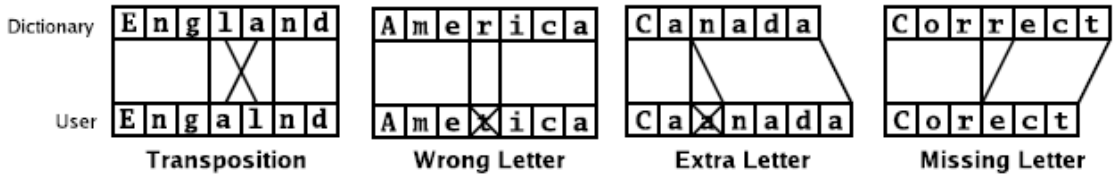

Figura 3-4: Os quatro mais comuns erros de ortografia.

Corretor Ortográfico no SCAE

Dentre as possíveis ferramentas de correção ortográfica, a escolhida para implementarmos o corretor ortográfico no SCAE foi a ferramenta *GNU Aspell* (ASPELL, 2008), pois essa utiliza técnicas de correção ortográfica que permitem corrigir erros fonéticos e propor melhores sugestões para as palavras escritas erradas. Além disso, a ferramenta trabalha com palavras em Português do Brasil.

O corretor ortográfico foi implementado como um módulo do DB_CORE, localizado no diretório DB_CORE/SPELLER. Como a ferramenta *Aspell* faz uso de um dicionário para propor uma lista de palavras corretas, disponibilizamos o dicionário no diretório DB_CORE/SPELLER/dictionary. Nesse diretório existem dois dicionários, dict_scae.rws e o pt_BR.rws. O primeiro foi gerado a partir do dicionário do NILC e as bases CNAE_SUBCLASSE_110 e DADOS_VITORIA_SUB_110, e o segundo foi obtido de (ASPELL, 2008). Nesse diretório existem outros arquivos que são de configuração para o *Aspell*.

O *Aspell* possui vários modos para sugerir possíveis palavras corretas dado uma errada: *ultra*, *fast*, *normal*, *slow*, *bad-spellers*. Em testes realizados, o método que apresentou melhores resultados e o que definimos como padrão para o corretor ortográfico foi o *slow*. Ainda, a ferramenta foi configurada para realizar análise de tipografia, passando um arquivo,

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

keyboard.kbd, com a configuração de um teclado ABNT2 em português, localizado no diretório DB_CORE/SPELLER/dictionary.

A partir de uma palavra errada, a ferramenta sugere várias palavras supostamente corretas em uma lista. A cada palavra sugerida, o *Aspell* atribui uma pontuação, ou *score*, onde a do topo da lista de *scores* é adotada como correta. Em testes realizados, percebemos que em muitas situações a palavra correta estava na lista de sugestão do *Aspell*, mas não se encontrava no topo.

Visando melhorar o desempenho do corretor ortográfico, utilizamos uma lista de palavras com as respectivas freqüências. O novo *score*, *rank*, é calculado a partir do *score* atribuído pelo *Aspell* e a freqüência da palavra na lista conforme a Equação 3-2. Então, a palavra considerada correta é que tiver o menor *rank*.

$$FrequencyScore = 1 + \ln(WordFrequency)$$

$$Rank = AspellScore / FrequencyScore$$

Equação 3-2: Cálculo do novo *score*, *rank*, para o corretor ortográfico.

Os códigos fonte do *Aspell* com as respectivas mudanças para atender a modificação anterior está no diretório DB_CORE/SPELLER/scaeaspell-0.60.5. A integração do DB_CORE com o *Aspell* é realizada por meio de bibliotecas dinâmicas, localizadas no diretório DB_CORE/SPELLER/scaeaspell-0.60.5/.libs, geradas durante a compilação do fonte do *Aspell*.

3.2 Meta Física 1.2/2007: Desenvolvimento de Mecanismo de Codificação Baseado em Redes Neurais Artificiais – Fundamentação do Código

Nesta seção, para conveniência do leitor, reapresentamos a descrição de Redes Neurais Sem Peso (RNSP) VG-RAM padrão e, em seguida, uma nova arquitetura de RNSP VG-RAM desenvolvida dentro do escopo deste Projeto – Redes Neurais Sem Peso VG-RAM com Correlação de Dados.

3.2.1 Redes Neurais Sem Peso VG-RAM

Redes neurais sem peso (RNSP), também conhecidas como redes neurais baseadas em *Random Access Memories* (RAM), não armazenam conhecimento em suas conexões, mas em memórias do tipo RAM dentro dos nodos da rede, ou neurônios. Estes neurônios operam com valores de entrada binários e usam RAM como tabelas-verdade: as sinapses de cada neurônio coletam um vetor de *bits* da entrada da rede, que é usado como o endereço da RAM, e o valor armazenado neste endereço é a saída do neurônio. O treinamento pode ser feito em um único passo e consiste basicamente em armazenar a saída desejada no endereço associado com o vetor de entrada do neurônio.

Apesar da sua notável simplicidade, as RNSP são muito efetivas como ferramentas de reconhecimento de padrões, oferecendo treinamento e teste rápidos, e fácil implementação. No entanto, se a entrada da rede for muito grande, o tamanho da memória dos neurônios da RNSP torna-se proibitivo, dado que tem de ser igual a 2^n, onde n é o tamanho da entrada. As redes *Virtual Generalizing* RAM (VG-RAM) são redes neurais baseadas em RAM que somente requerem capacidade de memória para armazenar os dados relacionados ao conjunto de treinamento.

Os neurônios RNSP VG-RAM armazenam os pares entrada-saída observados durante o treinamento, em vez de apenas a saída. Na fase de teste, as memórias dos neurônios VG-RAM são pesquisadas mediante a comparação entre a entrada apresentada à rede e todas as entradas nos pares entrada-saída aprendidos. A saída de cada neurônio VG-RAM é determinada pela saída do par cuja entrada é a mais próxima da entrada apresentada – a função de distância adotada pelos neurônios VG-RAM é a distância de *hamming*, i.e., o número de *bits* diferentes entre dois vetores de *bits* de igual tamanho. Se existir mais do que um par na mesma distância mínima da entrada apresentada, a saída do neurônio é escolhida aleatoriamente entre esses pares.

A Figura 3-5 ilustra a tabela-verdade de um neurônio VG-RAM com três sinapses (*X1*, *X2* e *X3*). Esta tabela-verdade contém três entradas (pares entrada-saída) que foram armazenadas durante a fase de treinamento (*entry* #1, *entry* #2 e *entry* #3). Durante a fase de teste, quando um vetor de entrada é apresentado à rede, o algoritmo de teste VG-RAM calcula a distância entre este vetor de entrada e cada entrada dos pares entrada-saída armazenados na tabela-verdade. No exemplo da Figura 3-5, a distância de *hamming* entre o vetor de entrada (*input*) e

a entrada #1 é dois, porque ambos os *bits X2* e *X3* não são semelhantes aos *bits X2* e *X3* do vetor de entrada. A distância da entrada #2 é um, porque *X1* é o único *bit* diferente. A distância da entrada #3 é três, como o leitor pode facilmente verificar. Portanto, para este vetor de entrada, o algoritmo avalia a saída do neurônio, *Y*, como "*category* 2", pois é o valor de saída armazenado na entrada #2.

lookup table	**X_1**	**X_2**	**X_3**	**Y**
entry #1	1	1	0	category 1
entry #2	0	0	1	category 2
entry #3	0	1	0	category 3
	↑	↑	↑	↓
input	1	0	1	**category 2**

Figura 3-5: Tabela-verdade de um neurônio da RNSP VG-RAM

Para classificar documentos de texto usando uma RNSP VG-RAM, um documento é representado por um vetor multidimensional $V = \{v_1, v_2, \ldots, v_{|V|}\}$, onde cada elemento v_i corresponde a um peso associado a um termo específico do vocabulário de interesse. Uma RNSP VG-RAM de uma única camada (veja Figura 3-6) é utilizada, de forma que as sinapses $X = \{x_1, x_2, \ldots, x_{|X|}\}$ de seus neurônios são conectadas aleatoriamente à entrada da rede $N = \{n_1, n_2, \ldots, n_{|N|}\}$, que tem o mesmo tamanho de um vetor que representa um documento, i.e., $|N| = |V|$. Note que $|X| < |V|$ (nossos experimentos demonstraram que $|X| < |V|$ provê melhor desempenho). Cada sinapse x_i de um neurônio forma uma célula *minchinton* com a próxima x_i+1 ($x_{|x|}$ forma uma célula *minchinton* com x_1). O tipo de célula *minchinton* usada retorna 1 se a sinapse x_i da célula é conectada a um elemento de entrada n_j cujo valor é maior do que aquele do elemento n_k ao qual a sinapse x_i+1 é conectada (i.e., $n_j > n_k$); caso contrário, ela retorna zero.

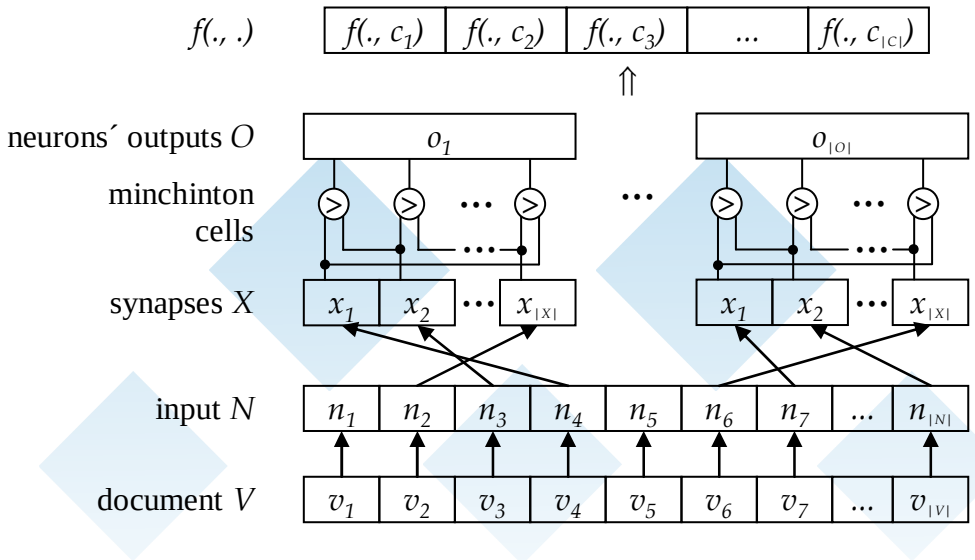

Figura 3-6: Arquitetura para classificação de texto da RNSP VG-RAM

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Durante o treinamento, para cada documento no conjunto de treinamento, o vetor correspondente V é conectado à entrada N da RNSP VG-RAM e as saídas $O = \{o_1, o_2, \ldots, o_{|o|}\}$ dos neurônios a uma das classes do documento. Todos os neurônios da RNSP VG-RAM são então treinados para retornar como saída esta classe com este vetor de entrada. O treinamento para este vetor de entrada é repetido para cada classe associada ao documento correspondente. Durante a fase de teste, para cada documento de teste, a entrada é conectada ao vetor correspondente e o número de neurônios retornando para cada classe é contado. A saída da rede é computada dividindo-se a contagem de cada classe pelo número de neurônios da rede.

A saída da rede é reorganizada como um vetor cujo tamanho é igual ao número de classes existentes. O valor de cada elemento deste vetor varia entre 0 e 1 e representa a porcentagem de neurônios que exibiram a classe correspondente como saída (a soma dos valores de todos os elementos deste vetor é sempre 1). Desta forma, a saída da rede reorganizada deste modo implementa a função $f(.,.)$, que apresenta valores no domínio dos números reais e que mapeia a múltipla pertinência de um documento frente a um dado conjunto de classes existentes. Finalmente, um valor limite τ_i para cada classe c_i pode ser usado com a função $f(.,.)$, a fim de definir o conjunto de classes a serem atribuídas a um documento de teste d_j: se $f(d_j, c_i) >= \tau_i$ então c_i é atribuída a d_j.

3.2.2 Redes Neurais Sem Peso VG-RAM com Correlação de Dados

Enquanto numa RNSP VG-RAM cada neurônio é treinado para retornar como saída uma única classe para cada vetor de entrada, numa RNSP VG-RAM com Correlação de Dados (RNSP VG-RAM-COR) cada neurônio pode ser treinado para retornar como saída um conjunto de classes para cada vetor de entrada. A Figura 3-7 ilustra a tabela-verdade de uma RNSP VG-RAM-COR com três sinapses *X1*, *X2* e *X3* e três entradas (pares entrada-saída) armazenadas durante a fase de treinamento (*entry* #1, *entry* #2 e *entry* #3). Semelhante à RNSP VG-RAM, quando um vetor de entrada é apresentado à rede na fase de teste, o algoritmo de teste da RNSP VG-RAM-COR computa a distância entre este vetor de entrada e cada entrada dos pares entrada-saída na tabela-verdade. No exemplo da Figura 3-7, a distância de *hamming* entre o vetor de entrada (input) e as entradas #1, #2, e #3 é dois, um e três, respectivamente. Como a entrada #2 da tabela-verdade é a mais próxima da entrada da rede, a saída do neurônio da RNSP VG-RAM-COR é dada pelas classes 1 e 3, i.e. o valor de *Y* representa ambas as classes, 1 e 3.

lookup table	**X_1**	**X_2**	**X_3**	**Y**
entry #1	1	1	0	category 2
entry #2	0	0	1	category 1, 3
entry #3	0	1	0	category 1, 2, 3
	↑	↑	↑	↓
input	1	0	1	**category 1, 3**

Figura 3-7: Tabela-verdade de um neurônio da RNSP VG-RAM-COR

Para classificar documentos de texto usando uma RNSP VG-RAM-COR, a mesma configuração da RNSP VG-RAM, ilustrada na Figura 3-7, é usada. Na fase de treinamento, para cada documento no conjunto de treinamento, o vetor correspondente *V* é conectado à entrada da RNSP VG-RAM-COR, *N*, e as saídas dos seus neurônios, *O*, ao conjunto de classes atribuído ao documento. Cada neurônio da RNSP VG-RAM-COR é treinado para retornar como saída este conjunto com este vetor de entrada. Durante a fase de teste, para cada documento de teste, o vetor correspondente *V* é conectado à entrada da rede, *N*. A função *f*(., .) é computada ao dividir o número de votos para cada classe pelo número total de classes retornadas pela rede. O número de votos para cada classe é obtido ao contar suas ocorrências em todos os conjuntos retornados pela rede.

A RNSP VG-RAM-COR foi implementada e adicionada ao SCAE, correspondendo ao core classificado WNN_COR_CORE.

3.3 Meta Física 1.3/2007: Desenvolvimento de Mecanismo de Codificação Baseado em Redes Bayesianas – Fundamentação do Código

A tarefa de atribuir um código CNAE a um dado documento de atividades tem uma semelhança com a tarefa de recuperação de documentos, onde se tenta localizar um documento em uma base de documentos. Na recuperação de documentos, um texto de busca (palavras-chave) é dado como entrada e a máquina de busca tenta localizar um documento que, diretamente ou indiretamente, relaciona-se com o texto dado. A maior diferença no nosso caso encontra-se no uso de classificações conhecidas na fase de treinamento, em vez de se referir simplesmente aos textos de uma base de documentos.

Nesta seção descrevemos uma abordagem para conduzir a tarefa de classificação, onde utilizamos uma ferramenta baseada no conceito de redes Bayesianas. Uma rede Bayesiana tenta modelar as dependências ao uso de palavras nos documentos atribuídos para cada subclasse de CNAE. Estas dependências são representadas sob a forma de uma rede de nós e arcos, onde cada nó corresponde a uma palavra e cada arco inclui informações sobre dependências entre as palavras em questão.

Nossa abordagem é baseada em um trabalho desenvolvido para um sistema de recuperação de documentos por Fernández-Luna (FERNÁNDEZ-LUNA, 2001) e De Campos *et al.* (DE CAMPOS, 2003).

Na seção a seguir apresentamos alguns conceitos básicos relacionados às redes Bayesianas. Depois mostramos como estes podem ser aplicados a nossa tarefa de classificação descrevendo em detalhe a estrutura de rede escolhida. Na Seção 3.3.3 descrevemos os passos seguidos na fase de treinamento em nossa formulação corrente da rede. O procedimento de classificação é apresentado na Seção 3.3.4. A implementação da ferramenta computacional é descrita na Seção 3.3.5. Os resultados obtidos com ajuda desta ferramenta são mostrados na Seção 3.3.7. As conclusões são o tópico da Seção 3.3.8.

3.3.1 Redes Bayesianas

Tecnicamente define-se uma rede Bayesiana *N* como tripleto *(V, A, P)*, onde:

1. *V* é um conjunto de nós (variáveis);
2. *A* é um conjunto de arcos, tal que, junto com *V*, temos um grafo acíclico *G = (V, A)*;
3. $P = \{P(v \mid \pi(v)) : v \in V\}$, onde $\pi(v)$ designa o conjunto de pais de nó *v*. O conjunto *P* consiste de probabilidades condicionais de todas as variáveis em relação a seus pais.

Tipicamente, os valores das variáveis podem ser valores booleanos, *T* (*true*) e *F* (*false*) (ou 1 e 0), com probabilidades determinadas por *P* e os arcos representam relações causais entre as variáveis.

Como exemplo, a Figura 3-8 mostra uma rede Bayesiana com quatro variáveis.

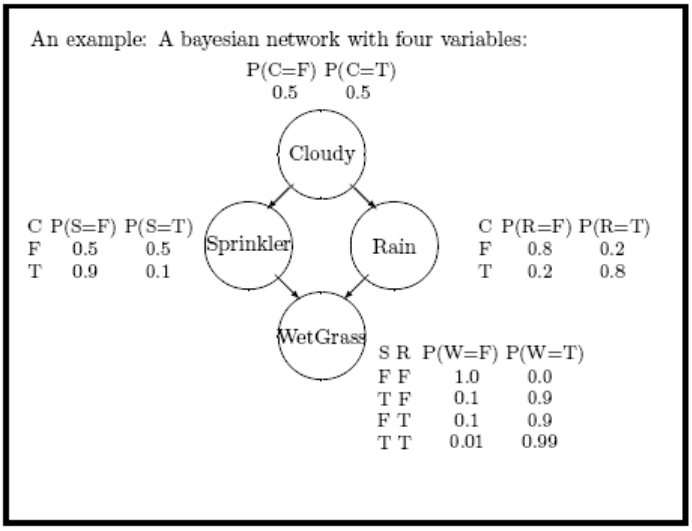

Figura 3-8: Uma rede Bayesiana.

No exemplo temos a variável *Cloudy* (nublado) com probabilidades iguais (0,5) de que o céu esteja nublado ou claro. A variável *Sprinkler* tem suas probabilidades associadas ao fato de que um sprinkler será utilizado se o céu estiver nublado ou não. Vemos que, se não estiver nublado, as probabilidades são iguais, mas, com tempo nublado, é pouco provável (probabilidade = 0.1) que o sprinkler seja utilizado. A variável *Rain* está associada às probabilidades de que teremos chuva se o céu estiver nublado ou não. Finalmente, a variável *WetGrass* contém as probabilidades de que a grama fique molhada nos casos em que um sprinkler seja utilizado ou uma chuva aconteça. Vê-se, por exemplo, nos casos extremos, em que nenhuma das duas condições acontece, que a probabilidade de se ter a grama seca é 1.0, e se as duas condições forem observadas a probabilidade de ter grama molhada é 0.99.

Uma rede Bayesiana pode ser considerada como um resumo de probabilidades conjuntas dos eventos denotadas em nosso exemplo por *P(C, S, R, W)*. Ou seja, uma abreviação para as probabilidades correspondendo a todas alternativas possíveis, ou seja, às probabilidades para todas as 16 combinações *P(C=T, S=T, R=T, W=T)*, *P(C=F, S=T, R=T, W=T)*, P(C=T, S=F, R=T, W=T),..., P(C=F, S=F, R=F, W=F).

Agora, a regra de cadeia de probabilidades permite que a expressão das probabilidades conjuntas dos eventos seja calculada a partir das probabilidades condicionais:

$$P(C, S, R, W) = P(C) \times P(C \mid S) \times P(R \mid C, S) \times P(W \mid C, S, R)$$

Equação 3-3

A idéia principal por traz do uso de uma rede Bayesiana é simplificar esta expressão tomando conta das dependências e independências entre as variáveis. Sabendo que as variáveis *S* e *R* (*Sprinkler* e *Rain*) são consideradas como independentes (é claro que se chover, vai chover

sem levar em consideração a decisão do dono da grama de usar seu sprinkler) podemos escrever:

$$P(R \mid C, S) = P(R \mid C)$$

Equação 3-4

Igualmente, a grama vai ser molhada somente como conseqüência de uso do sprinkler ou da ocorrência da chuva, o céu nublado não é uma causa direta. Assim, podemos simplificar a expressão correspondente tal que:

$$P(W \mid C, S, R) = P(W \mid S, R)$$

Equação 3-5

Assim, o cálculo da probabilidade conjunta *P(C, S, R, W)* reduz-se à expressão:

$$P(C, S, R, W) = P(C) \times P(C \mid S) \times P(R \mid C) \times P(W \mid S, R)$$

Equação 3-6

Temos como conseqüência prática que para o cálculo desta probabilidade seria suficiente guardar nas tabelas de probabilidades conjuntas associadas às variáveis do sistema somente aquelas mostradas na Figura 3-8.

Por que estamos interessados nesta redução no tamanho das tabelas? Podemos imaginar que queremos entender porque a grama está molhada. Será que foi por causa de uma chuva ou porque o dono da grama tinha ligado seu sprinkler? Vamos calcular as probabilidades para ambos os eventos. Para o primeiro:

$$P(R=T \mid W=T) = P(R=T, W=T) / P(W=T)$$

Equação 3-7

Esta é a famosa regra de Bayes. Uma probabilidade de um dado evento com condição é igual à probabilidade de todos os eventos comuns divida pela probabilidade do evento observado. Esta probabilidade pode ser computada aproveitando-se de conhecimento das probabilidades de eventos comuns, sendo dada neste caso por:

$$P(R = T|W = T) = \sum(P(C = c, S = s, R = T, W = T))/P(W = T),$$

Equação 3-8

onde a soma é calculada sobre os valores *T* e *F* das variáveis *C* e *S*. Em nosso exemplo este valor seria igual a 0,708. O mesmo cálculo daria para a probabilidade *P(S=T | W=T)* o valor 0,430. Assim, podemos duvidar que a causa de encontrar a grama molhada fosse à chuva. Vale destacar neste contexto que todas as conclusões probabilísticas incluem certo fator de incerteza. Uma maneira de expressar esta incerteza é calcular um fator de crença (*likelihood*),

expressado como razão das probabilidades obtidas. Em nosso exemplo, a crença seria 0.708/0.430 = 1.647. Quanto maior esta razão, o maior será a confiança que podemos ter em nossa decisão.

A estrutura de uma rede Bayesiana oferece diretamente meios para os cálculos práticos deste valores. Para calcular uma probabilidade atual em um nó, aproveita-se um procedimento recursivo, chamado propagação de crença (*belief propagation*). Sem entrar em mais detalhes técnicos deste procedimento, podemos resumir que se trata de um algoritmo proposto por Perl (PEARL, 1988) para passagem de mensagens entre os nós na rede Bayesiana de maneira que o valor de crença de um nó é sempre atualizado por aqueles valores associados a seus filhos e pais. Isto é implementado por um mecanismo onde cada nó envia mensagens para seus pais e para seus filhos, e atualiza seus valores de crença a partir das mensagens recebidas. O procedimento todo é sempre baseado nas tabelas de probabilidades condicionais associadas a cada nó.

3.3.2 Estrutura da Rede Bayesiana deste Trabalho

A rede Bayesiana construída para este trabalho constituiu-se de dois níveis de nós. O primeiro nível compõe-se de nós associados aos termos (palavras) presentes nos documentos de atividades para serem analisados. O segundo nível de nós corresponde às subclasses CNAE existentes. Formalmente, estamos notando com T_i os termos e com D_i as subclasses. A idéia principal é utilizarmos o algoritmo de propagação de mensagens somente no primeiro nível e atualizar os valores de crença no segundo aproveitando um cálculo de somas ponderadas baseadas nos valores de crença dos pais de cada nó deste nível.

Fazemos uma restrição: a estrutura da rede no primeiro nível tem que ser aquela de uma estrutura chamada poli-árvore, isto é, nenhum par de nós pode ter dois caminhos diferentes entre eles na estrutura da rede. É uma restrição prática e nosso objetivo futuro é relaxarmos tal restrição.

3.3.3 Fase de Treinamento

O treinamento de uma rede Bayesiana compreende duas tarefas principais: decidir qual será a estrutura da rede, e quais serão as probabilidades condicionais associadas aos nós da rede. Em detalhe, esta tarefa compõe-se dos seguintes passos:

1. Computação de graus de dependência entre todos os pares de nós.
2. Construção de um esqueleto, na forma de uma árvore, como uma estrutura inicial da rede.
3. Orientação de arcos da árvore para construir uma poli-árvore (a rede final).
4. Computação das probabilidades condicionais para cada nó da rede construída.

A seguir, todos estes passos são detalhados.

Passo 1 - Computação de graus de dependência

Um grau de dependência entre dois termos mede as relevâncias conjuntas dos mesmos para as subclasses de CNAE. O cálculo deste grau de dependência é baseado na tabela de freqüências dada como entrada na fase de construção da rede Bayesiana. Por exemplo, dados cinco termos, T_1, T_2, T_3, T_4 e T_5 e três subclasses de CNAE, SC_1, SC_2 e SC_3, poderíamos ter uma tabela de freqüências do tipo:

	T_1	T_2	T_3	T_4	T_5
SC_1	1	1	0	0	1
SC_2	0	1	0	1	0
SC_3	1	0	1	0	0

Figura 3-9: Uma tabela de freqüências.

Assim, o termo T_1, por exemplo, foi encontrado nas definições das subclasses SC_1 e SC_3 (o valor igual a 1), enquanto que o termo T_2 foi encontrado nas definições das subclasses SC_1 e SC_2. Os valores na tabela serão aproveitados para estimar as probabilidades das presenças e ausências dos termos nas definições das subclasses. Por exemplo, para o termo T_1, a probabilidade $P(T_1 = 1)$ é estimada como igual a 2/3. Estaremos também interessados nas probabilidades de presenças e ausências conjuntas dos termos nas definições, por exemplo, a probabilidade $P(T_1 = 1, T_2 = 1)$ é estimada como igual a 1/3, pois somente na primeira linha encontramos os dois com valor igual a um, e a probabilidade $P(T_1 = 1, T_2 = 0)$ é estimada como igual a 1/3, pois somente uma das linhas (a terceira) inclui esta combinação.

A definição formal de um grau de dependência entre dois termos T_i e T_j é dada por:

$$Dep(T_i, T_j|\emptyset) = \sum_{\mathbf{T_i},\mathbf{T_j}} p(\mathbf{T_i}, \mathbf{T_j}) ln(\frac{p(\mathbf{T_i}, \mathbf{T_j})}{p(\mathbf{T_i})p(\mathbf{T_j})}).$$

Equação 3-9

Vale destacar que um grau de dependência entre dois termos T_i e T_j mede não somente as presenças conjuntas deles nas definições das subclasses, mas também as coincidências de todas as combinações de presença, isto é, as ausências conjuntas e ainda as situações quando um e presente e o outro não estão presentes. Assim, no cálculo de um grau de dependência estamos somando as probabilidades de presenças e ausências conjuntas sobre todos os possíveis valores de variáveis randômicas T_i e T_j.

Passo 2 - Construção de uma árvore de esqueleto

O nosso objetivo final é construir uma rede Bayesiana que represente adequadamente todas as dependências entre os termos encontrados nas definições das subclasses de CNAE. Em uma primeira etapa construiremos uma estrutura que pode ser considerada como um esqueleto para a rede final. Neste estágio queremos primeiramente identificar as dependências fortes entre os termos.

Nossa abordagem, baseada na proposta do artigo de referência deste trabalho, é gerar uma árvore geradora máxima usando os graus de dependências como pesos dos arcos. Isto é, queremos construir uma estrutura de um grafo sem ciclos, onde todos os nós têm no mínimo um arco incidente e onde a soma dos pesos de todos os arcos e máxima. Tudo isto começa com uma árvore, que se compõe somente de um nó e sem nenhum arco. Depois se procura um arco ligando um novo nó à arvore com a maior dependência, inclui-se este na árvore, continuando até todos os nós estarem ligados no grafo com um arco. Tecnicamente, estamos aproveitando o algoritmo bem conhecido na literatura chamado de algoritmo de Prim (GIBBONS, 1985).

Na implementação do algoritmo foi utilizado um valor liminar para que os valores de dependências muito baixos sejam eliminados das considerações durante a construção da árvore.

A Figura 3-10 mostra um exemplo de uma árvore geradora máxima construída nesta fase.

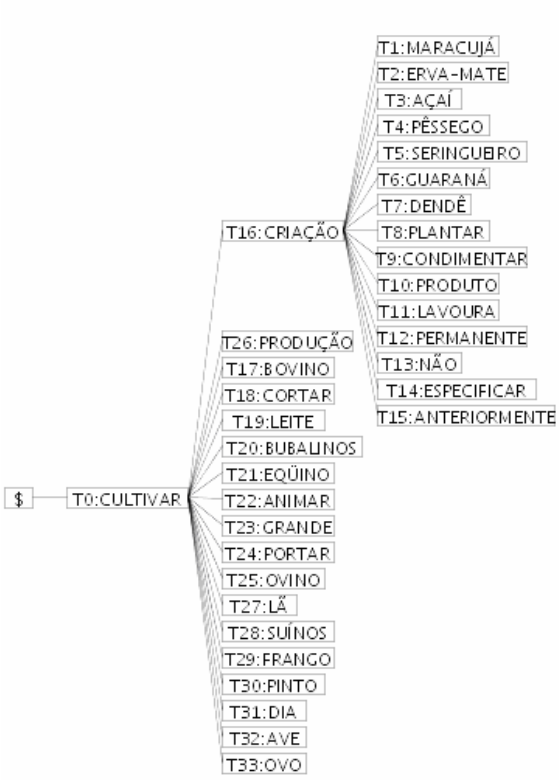

Figura 3-10: Uma árvore geradora.

Passo 3 – Orientação de arcos

Para gerarmos a estrutura final de uma rede Bayesiana, precisamos associar aos arcos de nosso esqueleto a sua direção. Uma questão principal é decidir para cada junção de três nós na árvore, digamos, para uma formada pelos nós T_i, T_j e T_k, se deveríamos direcionar arcos de T_i e T_j para o nó T_k ou se a direção deveria ser aquela saindo do nó T_k para os outros dois. A decisão tomada é baseada em uma comparação dos valores de dependência dos nós T_i e T_j levando em consideração o nó T_k, ou não. Nesta comparação, precisamos do conceito de

dependência tripla, que expressa as combinações de presença e ausência de termos associados aos nós T_i e T_j no caso que a presença (ou ausência) do nó T_k foi observada:

$$Dep(T_i, T_j|T_k) = \sum_{\mathbf{T_i},\mathbf{T_j},\mathbf{T_k}} p(\mathbf{T_i}, \mathbf{T_j}, \mathbf{T_k}) ln(\frac{p(\mathbf{T_i}, \mathbf{T_j}, \mathbf{T_k})p(\mathbf{T_k})}{p(\mathbf{T_i}, \mathbf{T_k})p(\mathbf{T_j}, \mathbf{T_k})}).$$

Equação 3-10

Podemos ver certa semelhança entre esta equação e a expressão da dependência dupla, isto é, sem observar o valor de T_k, especialmente se escrevemos:

$$\frac{p(\mathbf{T_i}, \mathbf{T_j}, \mathbf{T_k})p(\mathbf{T_k})}{p(\mathbf{T_i}, \mathbf{T_k})p(\mathbf{T_j}, \mathbf{T_k})} = \frac{p(\mathbf{T_i}, \mathbf{T_j}|\mathbf{T_k})}{p(\mathbf{T_i}|\mathbf{T_k})p(\mathbf{T_j}|\mathbf{T_k})}.$$

Equação 3-11

Isto é possível, pois se utilizarmos a definição de probabilidade condicional (Equação 3-7), obtemos a fórmula:

$$\frac{p(\mathbf{T_i}, \mathbf{T_j}, \mathbf{T_k})p(\mathbf{T_k})}{p(\mathbf{T_i}, \mathbf{T_k})p(\mathbf{T_j}, \mathbf{T_k})} = \frac{p(\mathbf{T_i}, \mathbf{T_j}|\mathbf{T_k})p(\mathbf{T_k})p(\mathbf{T_k})}{p(\mathbf{T_i}|\mathbf{T_k})p(\mathbf{T_k})p(\mathbf{T_j}|\mathbf{T_k})p(\mathbf{T_k})}$$

Equação 3-12

que logo podemos reduzir à Equação 3-11.

Baseado na comparação dos valores de dependência dupla *Dep*(T_i, T_j | Ø) e dependência tripla *Dep*(T_i, T_j | T_k), podemos fazer uma distinção entre os padrões $\mathbf{T_i} \rightarrow \mathbf{T_k} \leftarrow \mathbf{T_j}$ e $\mathbf{T_i} \leftarrow \mathbf{T_k} \rightarrow \mathbf{T_j}$ escolhendo a primeira, se observamos que o valor da dependência tripla é maior que o valor da dependência dupla, e a segunda no caso contrário.

Um exemplo da estrutura final da rede Bayesiana baseada na árvore geradora máxima do exemplo anterior pode ser visto na Figura 3-11. Os nós que correspondem às subclasses CNAE não são mostrados.

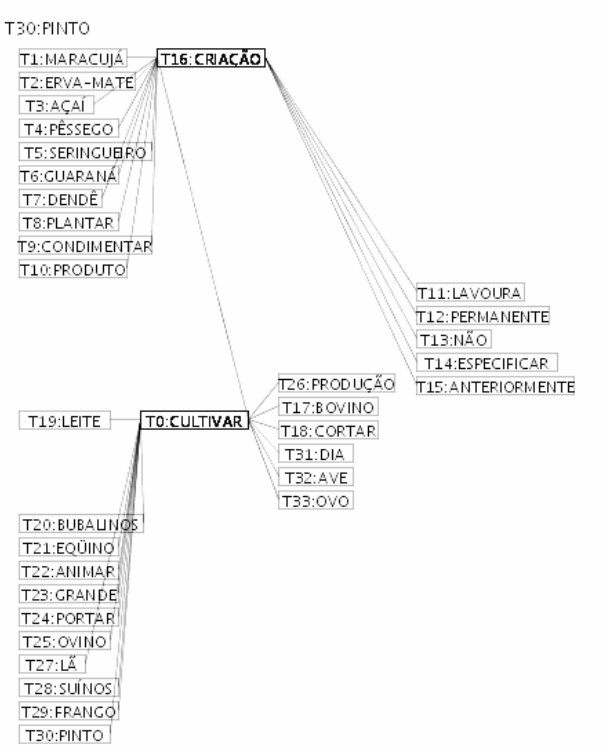

Figura 3-11: A estrutura final de uma rede Bayesiana (sem o nível das suclasses CNAE)

Passo 4 – Atribuir probabilidades condicionais para os termos

A cada termo T_i atribuímos uma tabela de probabilidades condicionais, onde as linhas correspondem às combinações de valores possíveis dos nós pais na rede e as colunas às probabilidades de que o termo T_i seja observado ou não, dada uma combinação de presenças e ausências dos pais. Nota-se por t_i o valor $T_i = 1$ e por $\bar{t}_i$ o valor $T_i = 0$.

Para um termo T_i sem nós pai, chamado nó raiz, define-se:

$$p(t_i) = 1/M \quad \text{e} \quad p(\bar{t}_i) = 1 - p(t_i),$$

Equação 3-13

onde M é o número total dos termos.

Para um termo T_i com pelo menos um nó pai define-se:

$$p(t_i|\pi(T_i)) = \frac{n(< t_i, \pi(T_i) >)}{n(\pi(T_i))}$$

Equação 3-14

e

$$p(\bar{t}_i|\pi(T_i)) = 1 - p(t_i|\pi(T_i)).$$

Equação 3-15

Nas fórmulas, $\pi(T_i)$ denota todos os nós pais do nó T_i e <..> indica uma das combinações e n<...> as freqüências da combinação nos dados de entrada calculadas da mesma maneira que no contexto do cálculo de dependências descrita acima.

Para nossa rede Bayesiana em dois níveis, definimos as probabilidades condicionais associadas aos nós de subclasses CNAE como uma soma dos pesos w_{ij}:

$$p(d_j|\pi(D_j)) = \sum_{T_i \in R_{\pi(D_j)}} w_{ij},$$

Equação 3-16

onde $R_{\pi(Dj)}$ é o conjunto de termos relevantes para uma subclasse da CNAE D_j. Os pesos w_{ij} definidos para cada termo T_i associado à subclasse de CNAE D_j são calculados por:

$$w_{ij} = \alpha^{-1} \frac{tf_{ij} \times idf_i^2}{\sqrt{\sum_{T_k \in R_{\pi(D_j)}} tf_{kj} \times idf_k^2}},$$

Equação 3-17

onde α é um coeficiente de normalização, tf_{ij} a freqüência do termo T_i relativo à subclasse D_j e

$$idf_i = log(\frac{N}{n_i}).$$

Equação 3-18

onde N é o número total de subclasses CNAE e n_i o número de subclasses CNAE tendo o termo T_i como termo relevante.

3.3.4 Fase de Classificação

Dado um documento descrevendo atividades econômicas, queremos encontrar as subclasses CNAE corretas para o mesmo. Nesta classificação temos os seguintes passos:

1. Instanciar cada termo T_i observado no documento, i.e., atribuir $p(t_i) = 1$;
2. Propagar a crença no nível dos termos utilizando o algoritmo exato de propagação chamado algoritmo de propagação de Pearl;
3. Calcular as probabilidades posteriores para todas as subclasses D_j da CNAE:

$$p(d_j|Q) = \sum_{T_i \in R_{\pi(D_j)}} w_{ij} p(t_i|Q).$$

Equação 3-19

4. Indicar como resultado as subclasses CNAE com os maiores valores $p(d_j|Q)$.

3.3.5 Ferramenta Computacional

O sistema categorizador baseado em Redes Bayesianas foi implementado em linguagem Java, que é orientada a objeto. Assim, o código do sistema foi organizado sob a forma de classes de objetos. Uma parte destas pode ser chamada classe de objetos de dados, pois a tarefa destas é providenciar uma representação dos elementos da rede Bayesiana.

Objetos deste tipo são:

Term e **DocClass** que são subclasses da classe **Node;**

ActivityDoc;

Dependency;

BayesNetwork.

Um segundo grupo de objetos são objetos “ativos”, encapsulando os procedimentos utilizados para construir uma rede Bayesiana:

NetworkBuilder;

PrimProcessor;

PolyTreeProcessor.

Adicionalmente precisamos de uma interface entre estes componentes e todos os demais componentes do sistema SCAE que foram implementados em linguagem C:

RPCHandler.

Esta interface transforma as chamadas do sistema SCAE para iniciar os métodos nativos escritos em Java.

3.3.7 Resultados dos Testes

A Figura 3-12 mostra os resultados preliminares obtidos com o categorizador baseado em redes Bayesianas.

Exp.	Dados de Treino			Dados de Teste				Rev. do SCAE	Des-emp.
	Tabela	Coluna	Limites	Tabela	Coluna	Limites	Nível		
4.1.1.1	110_SUBCL	DESCR_SUB	0 a 1182	110_SUBCL	DESCR_SUB	0 a 1182	SUBCL	613	98,82%
4.1.1.2	110_SUBCL	DESCR_SUB	0 a 1182	DADOS_VIX_110	OBJ_SOCIAL	0 a 3280	SUBCL	613	63,52%
4.1.1.3	110_SUBCL	DESCR_SUB	0 a 1182	DADOS_VIX_110	OBJ_SOCIAL	1640 a 3280	SUBCL	613	66,79%
4.1.1.4	110_SUBCL DADOS_VIX_110	DESCR_SUB OBJ_SOCIAL	0 a 1182 0 a 1639	DADOS_VIX_110	OBJ_SOCIAL	1640 a 3280	SUBCL	613	76,23%
4.1.1.5	110_SUBCL DADOS_VIX_110	DESCR_SUB OBJ_SOCIAL	0 a 1182 0 a 1639	DADOS_VIX_110	OBJ_SOCIAL	1640 a 3280	CLASSE	613	78,00%
4.1.1.6	110_SUBCL DADOS_VIX_110	DESCR_SUB OBJ_SOCIAL	0 a 1182 0 a 1639	DADOS_VIX_110	OBJ_SOCIAL	1640 a 3280	GRUPO	613	81,66%
4.1.1.7	110_SUBCL DADOS_VIX_110	DESCR_SUB OBJ_SOCIAL	0 a 1182 0 a 1639	DADOS_VIX_110	OBJ_SOCIAL	1640 a 3280	DIVISÃO	613	85,56%
4.1.1.8	110_SUBCL DADOS_VIX_110	DESCR_SUB OBJ_SOCIAL	0 a 1182 0 a 1639	DADOS_VIX_110	OBJ_SOCIAL	1640 a 3280	SEÇÃO	613	90,19%

Figura 3-12: Resultados dos testes

Observamos que melhores resultados foram obtidos com a inclusão de documentos descrevendo atividades econômicas no conjunto de treinamento (veja, por exemplo, testes 4.1.1.3 e 4.1.1.4). Correntemente, apenas documentos com uma única classificação foram utilizados.

3.3.8 Conclusão

Os resultados obtidos com o uso do sistema de categorização baseado em redes Bayesianas na categorização de atividades econômicas demonstram que esta é uma abordagem promissora. Uma nova versão do sistema poderia incluir os seguintes aprimoramentos:

1. Utilização de um algoritmo de busca para melhorar o direcionamento dos arcos da rede.
2. Utilização de um outro método de propagação de crenças na rede como, por exemplo, o método chamado *Noisy OR-Gate Method* (PEARL 1988). O objetivo seria evitar a explosão combinatória eventualmente causada por um grande número de nós pais associados a um nó.
3. Utilização de nós intermediários para diminuir o número de nós pais de um nó.

3.4 Meta Física 1.4/2007 – Desenvolvimento de Mecanismo de Codificação Baseado em Latent Semantic Indexing – Fundamentação do Código

Nesta seção, para conveniência do leitor, reapresentamos a descrição do modelo vetorial de classificação de documento baseada , em seguida, uma nova arquitetura de RNSP VG-RAM desenvolvida dentro do escopo deste Projeto – Redes Neurais Sem Peso VG-RAM com correlação de dados.

3.4.1 Representação Vetorial de Documentos

No modelo que adotamos neste trabalho, o vetorial, os documentos são representados por vetores no espaço R^n (BAEZA-YATES; RIBIERO-NETO, 1998), onde n representa o número de termos-palavras nos documentos considerados. Cada documento é considerado, portanto, um vetor de termos. Formalizando o que foi dito, consideremos um conjunto de documentos $D = \{d_1, d_2, ..., d_i, ...d_n\}$, onde d_i é um dos elementos desse conjunto. O documento d_i será representado, portanto, por um vetor de pesos $d_i = [w_1, w_2, ..., w_k, w_{k+1}, w_{k+2}, ...w_n]$, sendo que k é o número de todos os termos $\{t_1, t_2, ..., t_k\}$ distintos que aparecem no documento d_i. Os demais termos $\{t_{k+1}, t_{k+2}, ..., t_n\}$, associados aos pesos $\{..., w_{k+1}, w_{k+2}, ..., w_n\}$, são termos que aparecem em outros documentos. Portanto, $\{t_1, t_2, ...t_k, t_{k+1}, t_{k+2}, ..., t_n\}$ são todos os termos do vetor associado ao documento d_i e a freqüência dos termos $t_{k+1} = t_{k+2} = ... = t_n = 0$ nesse vetor.

Assim, podemos concluir que um termo (palavra no documento) pode aparecer em mais de um documento. Portanto, a cada termo será atribuído um peso w_i. O peso que esse termo recebe leva em consideração dois aspectos: a quantidade de vezes que ele ocorre no próprio documento e a quantidade de vezes que ele aparece em outros documentos analisados. Através disso, ponderamos a importância desse termo no conjunto de documentos onde ele aparece. Uma das propostas de ponderação dessa importância apresentada na literatura (BAEZA-YATES; RIBIERO-NETO, 1998) é dada pela equação:

$$idf_i = log(N/n_i)$$

Equação 3-20

onde idf_i, isto é, *inverse document frequency*, é o valor dessa ponderação para o termo t_i, N é o total de documentos no conjunto D e n_i, o número de documentos em que o termo t_i aparece. Por meio desta formulação, queremos reforçar o fato de que, se um termo aparece em todos os documentos, seu peso deve ser próximo de zero (idf_i assumirá um valor próximo de zero).

Para dar uma ilustração do que acabamos de formalizar, vejamos o exemplo dos procedimentos de construção do vetor representativo do documento dado a seguir.

Índice i	Peso w_i	Termo t_i d_1
1	3	campeonato
2	1	brasileiro
3	1	próximo
4	1	fim
5	1	foi
6	1	prejudicado
7	1	desorganização
8	2	times
9	1	famosos
10	1	poderão
11	1	rebaixados
12	1	entrando
13	1	justiça
14	1	pedir
15	1	anulação

Figura 3-13: Representação vetorial de um documento.

Considere que tenhamos a seguinte notícia na área de esporte: d_1: "O campeonato brasileiro está próximo ao fim. Tal campeonato foi muito prejudicado pela desorganização e times famosos poderão ser rebaixados. Alguns times estão entrando na Justiça para pedir a anulação do campeonato."

Primeiramente devemos excluir as palavras sem muito significado: os artigos e preposições, por exemplo – são as *stopwords* (BAEZA-YATES; RIBIERO-NETO, 1998). Ficaremos com a lista de palavras apresentada na Figura 3-13 quando analisarmos o documento d_1. Para facilitar o entendimento, nesse exemplo, consideraremos a influência dos $idf = 1$ para todos os pesos dos termos. Outra estratégia que estaremos adotando neste trabalho será utilizar, na representação vetorial do documento, apenas as palavras que tiverem peso maior que 50% do termo de maior peso. No caso da Figura 3-13, o termo de maior peso é a palavra campeonato, com peso 3. Assim, somente utilizaremos as palavras com peso igual ou superior a $3/2 = 1{,}5$. Com isso, ficamos somente com campeonato e times para a representação vetorial desse documento.

Agora, considere outros dois documentos que, depois do procedimento que acabamos de descrever, teriam os seguintes termos representativos:

- 1 : peso 5 para o termo campeonato, 4 para brasileiro e 3 para times;
- 2 : peso 2 para o termo campeonato, 3 para brasileiro e 1 para times;

Através desse exemplo ilustrativo e sua representação, é possível, agora, visualizar os três documentos de forma gráfica. Na forma gráfica, podemos ver a relação de distância que existe entre os documentos quando olhamos o ângulo que um vetor tem com o outro. Esse conceito de distância será muito utilizado mais adiante.

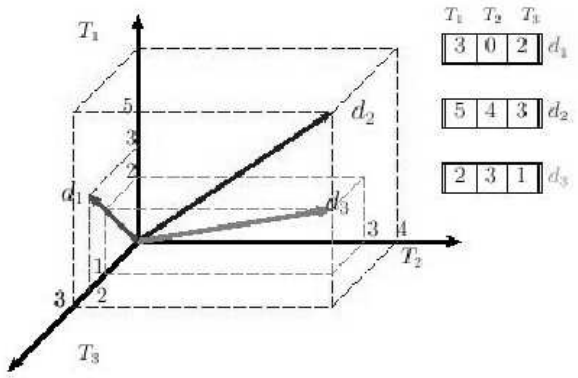

Figura 3-14: Representação gráfica de três vetores de acordo com o modelo vetorial

Na Figura 3-14, apresentamos a representação vetorial, de forma gráfica, de três documentos ilustrativos dessa metodologia. Os eixos $T_1 = campeonato$, $T_2 = brasileiro$ e $T_3 = times$ representam a magnitude dos pesos dos termos que aparecem nos documentos d_1, d_2 e d_3. O peso dado ao termo t_1 no documento d_2 foi 5, enquanto em d_3 foi 2, o que significa que esse termo tem uma importância maior para o segundo documento. Notamos que o termo t_2 não ocorre em d_1, por isso, está com valor nulo na segunda posição do vetor representativo do documento.

Essa forma de representar um documento nos mostra que enquanto nós, seres humanos, pensamos, as máquinas fazem contas. Portanto, o que está por trás de um modelo como esse é o fato de transformar o processo de indexação e classificação em um processo de contagem, para que o computador possa nos auxiliar a tratar grandes volumes de documentos.

Dessa forma, consideraremos a pequena base ilustrativa $D = \{d_1, d_2, d_3\}$ de documentos. O que queremos agora é saber, precisamente, quão similar é um documento ao outro. O que desejamos é calcular o valor de $sim(d_i, d_j)$ entre quaisquer dois documentos da base. Uma vez que temos a representação vetorial dos documentos da base, como apresentado na Figura 3-14, a conta que agora devemos fazer é a seguinte (BAEZA-YATES; RIBIERO-NETO, 1998):

$$sim(d_i, d_j) = \frac{\mathbf{d_i} \bullet \mathbf{d_j}}{|\mathbf{d_i}| \times |\mathbf{d_j}|} =$$

$$= \frac{\sum_{k=1}^{n} w_k^i \times w_k^j}{\sqrt{\sum_{k=1}^{n} \{w_k^i\}^2} \times \sqrt{\sum_{k=1}^{n} \{w_k^j\}^2}} = cos(\theta),$$

Equação 3-21

onde $|d_i|$ é o módulo do vetor d_i e $\cos(\theta)$ é o cosseno do ângulo entre os vetores que representam os dois documentos d_i e d_j. O valor do cosseno de um ângulo varia em um intervalo de 0 a 1. Esse fato nos dará uma interpretação de distância entre os documentos, onde 0 significará o mais alto grau de dissimilaridade e 1, completa similaridade. Já o valor w_k^i indica o peso referente ao termo no documento, como descrito anteriormente.

Vamos exemplificar utilizando os três documentos ilustrativos. Para os documentos d_1 e d_2, a conta é a seguinte:

$$sim(d_1, d_2) = \frac{3 \times 5 + 0 \times 4 + 2 \times 3}{\sqrt{3^2 + 0^2 + 2^2} \times \sqrt{5^2 + 4^2 + 3^2}} = \frac{21}{25.49} = 0.82 = cos(\theta_{1,2})$$

$$sim(d_1, d_3) = \frac{3 \times 2 + 0 \times 3 + 2 \times 1}{\sqrt{3^2 + 0^2 + 2^2} \times \sqrt{2^2 + 3^2 + 1^2}} = \frac{8}{13.49} = 0.59 = cos(\theta_{1,3})$$

$$sim(d_2, d_3) = \frac{5 \times 2 + 4 \times 3 + 3 \times 1}{\sqrt{5^2 + 4^2 + 3^2} \times \sqrt{2^2 + 3^2 + 1^2}} = \frac{25}{24.49} = 0.94 = cos(\theta_{2,3})$$

As contas realizadas indicam-nos que os documentos d_2 e d_3 têm o mais alto grau de similaridade entre os três documentos: 0.94. Note que, intuitivamente, podemos visualizar esse resultado no gráfico da Figura 3-14.

3.4.2 Avaliação do Desempenho do Algoritmo ML-kNN em Classificação de Textos de Atividades Econômicas

A classificação automática de textos é em geral um problema desafiador na literatura. Essa classificação pode ser aplicada a bases de dados que possuem duas características distintas: uma onde os documentos são classificados em uma única categoria, e outra onde os documentos podem ser classificados em um número indeterminado de categorias.

Nós investigamos o segundo tipo, também chamado de classificação multi-rotulada. Por ter obtido resultados superiores a outros algoritmos propostos para resolução desse mesmo tipo de problema (ZHANG; ZHOU, 2007), o algoritmo *Multi Label k-Nearest Neighbor (ML-kNN)* foi selecionado para ser aplicado ao problema proposto. No entanto, diferentemente do que foi feito em experimentos reportados na literatura (ZHANG; ZHOU, 2007), onde a quantidade máxima de categorias foi de 40 categorias, neste projeto aplicamos tal algoritmo a

uma base de dados com centenas de categorias. Assim, avaliamos como o algoritmo *ML-kNN* se comporta num domínio onde exista um elevado número de categorias possíveis de serem atribuídas a cada documento.

A característica de uma base de dados multi-rotulada cujos documentos podem estar relacionados a uma grande quantidade de categorias é encontrada em conjuntos de documentos que representam descrições de atividades econômicas de empresas. Para cada descrição das atividades econômicas de uma empresa é associada uma ou mais categorias de acordo com as definições das atividades pré-definidas na tabela Classificação Nacional de Atividades Econômicas (CNAE) (IBGE, 2003).

A tabela CNAE é definida em 5 níveis distintos: seção, divisão, grupo, classe e subclasse, nessa ordem. Neste estudo, os documentos foram classificados de acordo com o último nível, o qual possui 1183 sublcasses (CNAE 1.1), o que é incomum na literatura (SEBASTIANI, 2002). Existe uma grande demanda de classificação anual (aproximadamente 1,5 milhões) (DNRC, 2008), pois muitas novas empresas são abertas ou alteram suas atividades econômicas ao longo de cada ano.

Esta investigação está organizada na seguinte estrutura. Na Seção 3.4.2.1 iremos detalhar o funcionamento do algoritmo *ML-kNN*. Na Seção 3.4.2.2 será comentado quais foram os experimentos realizados e os resultados obtidos. Por fim, na Seção 3.4.2.3, a conclusão da investigação é apresentada.

3.4.2.1 Algoritmo ML-kNN

O *ML-kNN* é um classificador multi-label baseado no popular método *kNN* (*k-Nearest Neigbours*) (ZHANG; ZHOU, 2007).

Para cada documento de teste d_j, o *ML-kNN* inicialmente encontra seus k vizinhos mais próximos no conjunto de treino usado, isto é, encontra os k primeiros elementos ordenados pelo valor de similaridade com d_j de forma decrescente, usando, por exemplo, a distância Euclidiana. Mais tarde, o algoritmo identifica quantos exemplares de cada categoria existem dentre os k vizinhos mais próximos de d_j, que chamaremos de k_i ($i \in \{1, 2, ..., |C|\}$), onde $|C|$ é o número de categorias. Seja H_1^i o evento em que d_j possui o rótulo i e H_0^i o evento em que d_j não possui o rótulo i. E mais, seja E_j^i o evento em que existem j vizinhos mais próximos de d_j pertencentes à categoria i. Assim, temos:

$$y_{d_j}(i) = argmax_{b \in 0,1} P(H_b^i) P(E_j^i | H_b^i)$$

Equação 3-22

Na Equação 3-22, $y_{d_j}(i)$ é a probabilidade do documento d_j pertencer à categoria i. $P(H_b^i)$, (onde $i \in \{1, 2, ..., |C|\}$ e $b \in \{0, 1\}$) e $P(E_j^i | H_b^i)$ ($j \in \{0, 1, ..., k\}$) são, respectivamente, as probabilidades *a priori* e *a posteriori* da categoria i. Essas probabilidades são estimadas na etapa de treino do algoritmo *ML-kNN*, sendo que, primeiramente, é estimada a probabilidade *a priori* de cada categoria usando a seguinte equação:

$$P(H_1^i) = \frac{\delta + N_i}{2\delta + N} \qquad P(H_0^i) = 1 - P(H_1^i)$$

Equação 3-23

onde N_i denota o número de exemplares da categoria i no conjunto de treino e N denota o número total de exemplares. δ é um parâmetro para suavizar a probabilidade.

Após estimada a probabilidade *a priori*, o *ML-kNN*, para cada exemplar w_y no conjunto de treino (onde $y \in \{1, 2, ..., N\}$), encontra seus k vizinhos mais próximos e calcula o número total de votos que cada categoria recebe dos k vizinhos mais próximos. Em outras palavras, seja k_i o número de votos que cada categoria i recebeu de w_y, se o exemplar w_y pertence à categoria i, então será adicionado 1 a $L^i_{k_i}$, senão será adicionado 1 a $\overline{L^i_{k_i}}$. $L^i_{k_i}$ e $\overline{L^i_{k_i}}$ indicam quantos exemplares de treino estão relacionados com a categoria i, e respectivamente, não relacionados com a categoria i. Finalmente, com essas informações, as probabilidades *a posteriori* são calculadas como descrito na Equação 3-24 e na Equação 3-25:

$$P(E_j^i|H_1^i) = \frac{\delta + L_j^i}{\delta(k+1) + \sum_{o=0}^{k} L_o^i}$$

Equação 3-24

$$P(E_j^i|H_0^i) = \frac{\delta + \overline{L_j^i}}{\delta(k+1) + \sum_{o=0}^{k} \overline{L_o^i}}$$

Equação 3-25

O *ML-kNN* precisa apenas de dois parâmetros: o número k de vizinhos mais próximos e a suavização δ da probabilidade. Na Equação 3-23, na Equação 3-24 e na Equação 3-25 o valor de δ modifica levemente as probabilidades *a priori* e as probabilidades *a posteriori*.

3.4.2.2 Experimentos e Resultados

A base de dados utilizada no experimento é composta de 3281 documentos, que representam descrições de atividades econômicas de empresas da cidade de Vitória-ES, e por 1183 definições de subclasse da tabela CNAE. Foi realizado um pré-processamento desta base de dados antes que os experimentos fossem realizados. Neste pré-processamento, cada documento da base de dados foi submetido a (i) um processo de *stemming*, fazendo com que as palavras nos documentos ficassem sem gênero, número e grau, e a (ii) um processo de retirada de *stop words*, isto é, retirada de artigos, preposições, conjunções, números e outras palavras que apenas prejudicariam a caracterização do documento. Após o pré-processamento, e como feito por Oliveira et. al em (OLIVEIRA, 2007), cada documento foi

representado como um vetor no espaço R^n, onde n é o número total de termos encontrados no conjunto de documentos.

Os experimentos foram divididos em duas etapas: a etapa de validação e a etapa de teste. Na etapa de validação foram utilizadas 1183 definições de subclasse juntamente com 820 descrições de atividades econômicas no treinamento do algoritmo. Após o treinamento, a validação do algoritmo (obtenção do melhor valor do parâmetro *k*) foi realizada utilizando outras 820 descrições de atividades econômicas. A seguir, foi realizada a etapa de teste, onde o *ML-kNN* foi treinado com 1183 definições de subclasses e com 1640 documentos, e foi testado com outros 1641 documentos.

Os resultados foram analisados utilizando as métricas *Coverage*, *One Error*, *Average Precision* e *Ranking Loss*, definidas em (ZHANG; ZHOU, 2007). As métricas *One Error*, *Ranking Loss* e *Average Precision* são definidas no intervalo de 0 a 1. A métrica *Coverage* possui limite inferior igual a 0 e limite superior igual a $|C|$ - 1; no entanto, para uma melhor representação gráfica, essa métrica foi normalizada na faixa de 0 a 1. Para o entendimento dos resultados obtidos é importante destacar que quanto menor o valor encontrado para as métricas *Coverage*, *One Error* e *Ranking Loss* melhor é o resultado. Já em relação a métrica *Average Precision*, quanto maior o valor melhor o resultado. Os resultados obtidos são mostrados a seguir, na Figura 3-15.

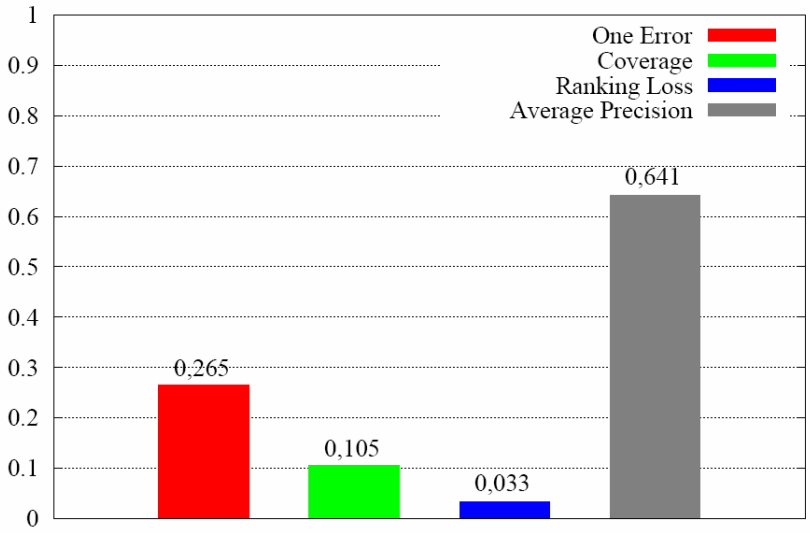

Figura 3-15: Resultados experimentais obtidos com o ML-kNN

Considerando o grande número de categorias presentes na base de dados, foram obtidos resultados expressivos nas métricas de desempenho utilizadas. Analisando os resultados obtidos e comparando com o resultado apresentado em (OLIVEIRA, 2007), muito embora as métricas utilizadas sejam diferentes, intuitivamente percebe-se que o *ML-kNN* retornou resultados melhores que o algoritmo *Vizinho Mais Próximo*.

3.4.2.3 Conclusão

Esta investigação teve como objetivo avaliar o desempenho do algoritmo *ML-kNN* quando empregado na classificação de uma base de dados com uma grande quantidade de categorias,

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

e foi constatado que ele pode ser aplicado à resolução de tal tipo de problema. Como trabalhos futuros são previstos realizar não apenas a comparação rigorosa entre o algoritmo *ML-kNN* e o algoritmo *Vizinho Mais Próximo*, mas também fazer comparações do algoritmo *ML-kNN* com outros algoritmos de classificação multi-rotulada. Um estudo que também é muito pertinente diz respeito a incorporar ao *ML-kNN* algumas técnicas para viabilizar a classificação de uma base de dados com um grande número de documentos.

3.5 Meta Física 1.5/2007: Desenvolvimento de Mecanismo de Composição dos Resultados da Codificação Através de Redes Neurais Artificiais, Redes Bayesianas e *Latent Semantinc Indexing* em uma Única Codificação, mais Robusta – Fundamentação do Código

Em seções anteriores foram expostas diferentes metodologias para a classificação de documentos que descrevem atividades econômicas:

- Redes Neurais Sem Peso (RNSPs);
- Latent Semantic Indexing (Modelo Vetorial);
- Redes Bayesianas.

A seguir, serão apresentados mecanismos para compor essas metodologias, utilizando um subconjunto qualquer delas para formar um novo classificador.

Além dos núcleos, ou cores, já implementados no SCAE, outros mecanismos de classificação poderão fazer parte do core ENSEMBLE – como foi denominado o núcleo de combinação – à medida em que forem incorporados ao sistema. Para tal, é necessário apenas que eles implementem a interface de comunicação entre os núcleos, definida pelas funções descritas na biblioteca *shared* do SCAE (mais detalhes sobre o SCAE e seus núcleos na Seção 3.6):

- begin_training e begin_test – inicializando as variáveis necessárias;
- train_line e test_line – implementando o procedimento para treinamento e teste de um objeto;
- end_training e end_test – desalocando variáveis e estruturas utilizadas durante o treinamento ou teste;
- classify – implementando o procedimento para classificar um objeto pelo interface web.

A Figura 3-16 ilustra a inclusão de um novo núcleo que implementa, por exemplo, uma *Support Vector Machine* (SVM).

O treinamento dos núcleos subordinados, apesar de poder ser invocado a partir do ENSEMBLE, permanece independente dele. É possível até mesmo que as bases de treinamento dos núcleos subordinados sejam atualizadas uma a uma, sem que isto prejudique o funcionamento do ENSEMBLE como um todo. A Figura 3-17 mostra a interligação do ENSEMBLE com outros núcleos.

Para realizar uma classificação, o ENSEMBLE recebe um dado texto descrevendo atividades econômicas (um objeto social, por exemplo) a ser classificado através da interface do SCAE e repassa-o para os núcleos subordinados. Cada um deles informa ao ENSEMBLE uma classificação que não é necessariamente composta por apenas uma classe. De posse destas informações, ou crenças dos classificadores a respeito da pertinência da classificação do

objeto social dentro de uma ou mais subclasses CNAE, o ENSEMBLE computa uma única classificação, que reflete as contribuições dos núcleos subordinados, e retorna-a para o usuário. Este processo está ilustrado na Figura 3-18.

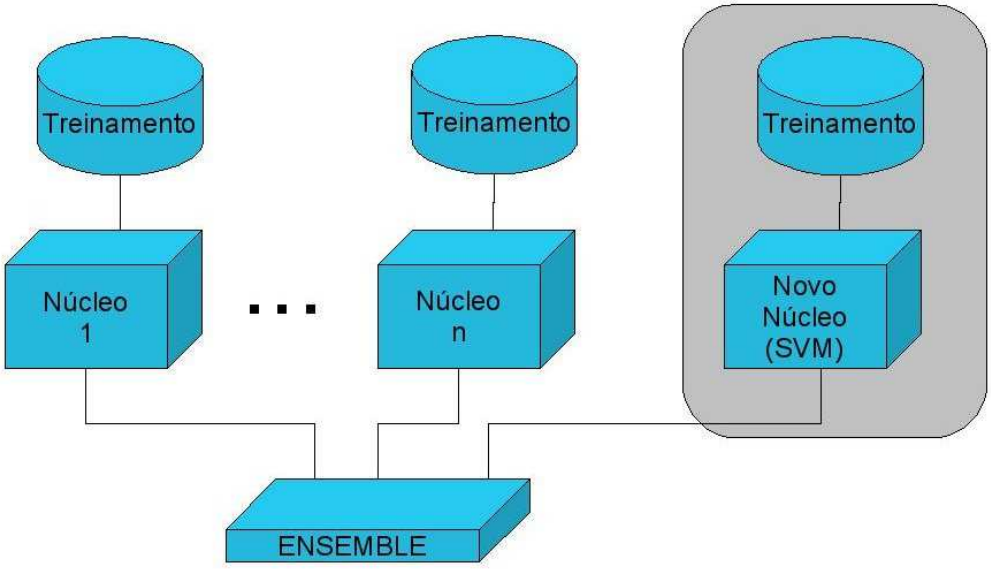

Figura 3-16: Inclusão de núcleos no SCAE

Há diversas alternativas de combinação das crenças dos núcleos subordinados em uma única classificação. Estas podem, grosseiramente, ser agrupadas naquelas que combinam de forma estática ou dinâmica.

3.5.1 Combinação Estática

Em sistemas que combinam classificadores estaticamente, as combinações são determinadas por uma análise estatística rigorosa, norteada pelo estudo de base de dados de classificações realizadas por classificadores humanos calibrados, isto é, cuja taxa de acerto é conhecida. A seção correspondente à Meta Física 3.2/2007 deste Relato (Seção 3.8.3) aprofunda este assunto ao discutir métodos para a medição de concordância entre classificadores, como o índice Kappa.

Outra possibilidade para combinar crenças estaticamente é Boosting, uma máquina de aprendizado supervisionado. O Boosting é baseado na questão proposta por Kearns (KEARNS, 1988): "*Can a set of weak learners create a single strong learner?*" Para resolver esta questão, muitos algoritmos de Boosting foram propostos. Os primeiros, cujas autorias atribuem-se a Schapire (SCHAPIRE, 1990) e Freund (FREUND, 1990), não são adaptativos, no sentido de que não alteram os pesos de cada classificador. Por isso, esses algoritmos não exploram totalmente o potencial dos classificadores. Originalmente, o Boosting foi empregado com classificadores fracos, que são aqueles que apresentam, individualmente, taxas de acerto pouco superiores a 50%.

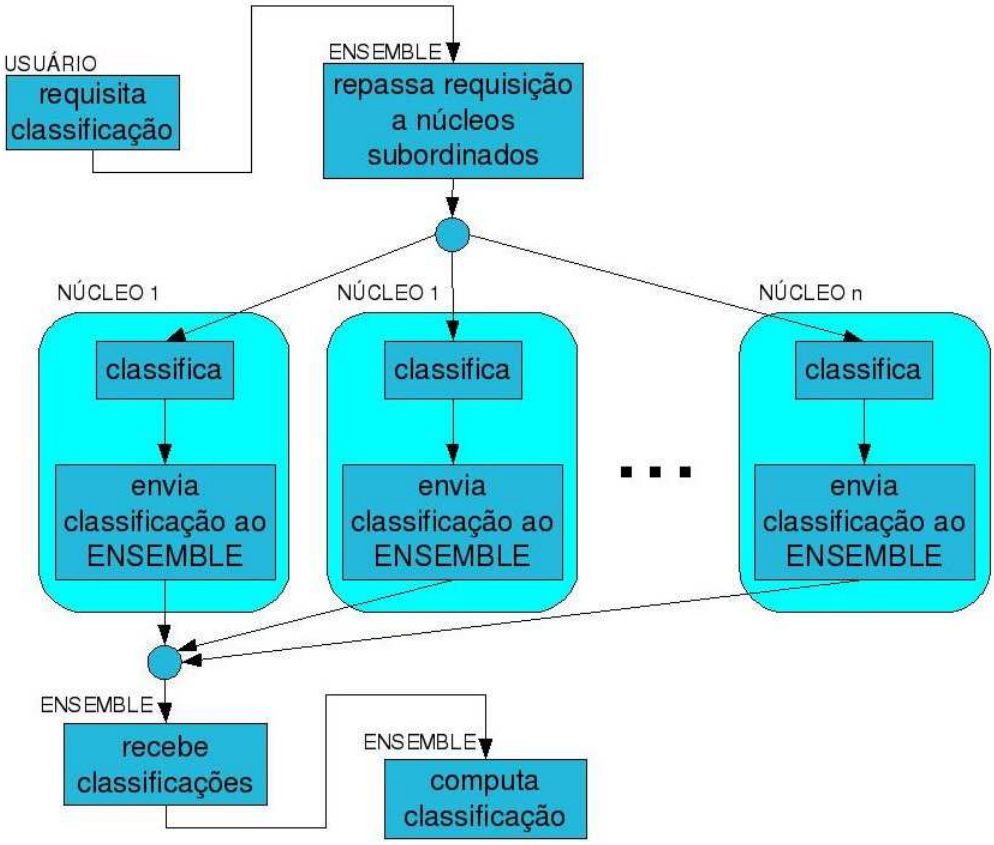

Figura 3-17: Interligação do ENSEMBLE com outros núcleos do SCAE

O algoritmo AdaBoost, introduzido em 1996 (FREUND; SCHAPIRE, 1996), é uma modificação do Boosting. Em contraste com o Boosting, no AdaBoost os pesos de cada um dos *n* classificadores que compõem o classificador final são obtidos de forma adaptativa. Durante o treinamento, cada um dos classificadores fracos recebe os padrões de entrada. A função de distribuição (que inicialmente é uniforme) é alterada usando-se uma fração do menor erro obtido dentre os *n* classificadores. Assim, ao final do treinamento, uma função de distribuição que pondera cada classificador é encontrada, e a soma ponderada de suas respostas que é a resultante do classificador final.

Também é possível combinar classificadores em cascata, utilizando o Boosting para o treinamento. Dentre os classificadores disponíveis, *n* são escolhidos. O objeto a ser classificado é então apresentado ao primeiro deles, que descarta algumas classificações. Esta informação é passada ao segundo classificador que, por sua vez, elimina mais algumas possibilidades. O procedimento continua até o *n*-ésimo classificador, o mais seletivo deles, que realiza a classificação sobre um conjunto já bastante reduzido de classes.

Uma das primeiras e mais conhecidas aplicações de um classificador em cascata é a detecção de faces em imagens. Esta aplicação, desenvolvida por Viola e Jones (VIOLA; JONES, 2001), tem bons resultados independentemente das dimensões da imagem.

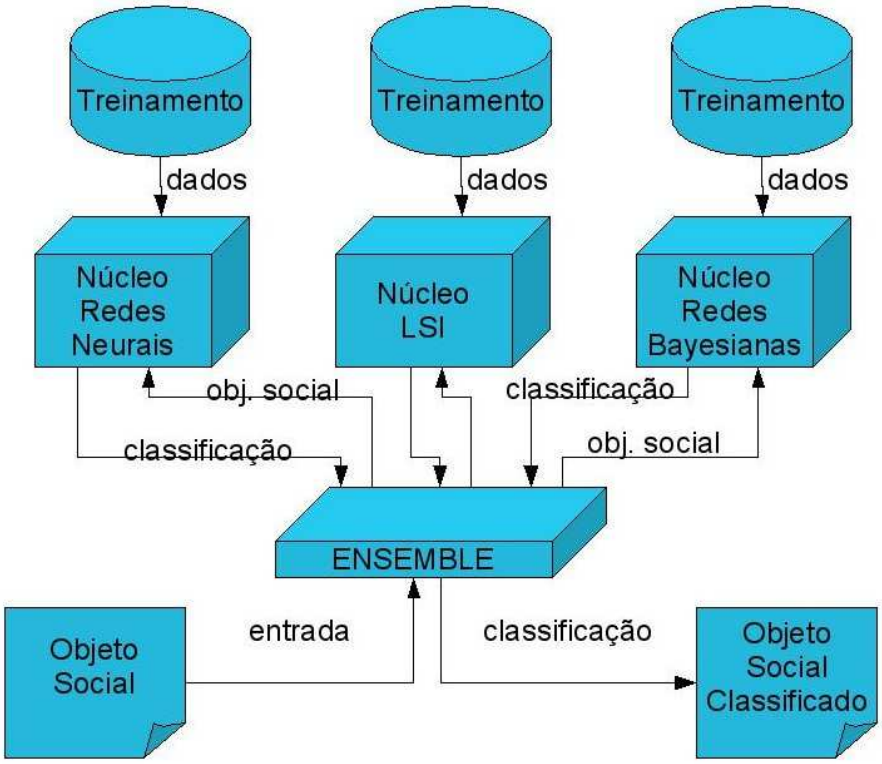

Figura 3-18: Classificação pelo ENSEMBLE

3.5.2 Combinação Dinâmica

Em sistemas que combinam classificadores dinamicamente, os pesos associados a cada classificador subordinado são determinados por uma análise do comportamento dos classificadores durante o processo de treinamento. No caso do SCAE, a dinamicidade dos pesos dos classificadores poderá ser obtida por meio de um segundo nível de aprendizado de máquina. Este seria realizado a partir de uma realimentação introduzida no sistema, que poderá vir de duas formas:

- como questionamento à classificação de um determinado objeto social (realimentação negativa);
- pela repetição de solicitações para a classificação de um objeto sem questionamento.

A realimentação negativa relacionada à eventual necessidade de uma reclassificação manual causaria a diminuição do peso dos classificadores votantes na categoria questionada. Já a realimentação positiva ocorreria em duas situações: (i) nos casos de reclassificação manual, através do aumento significativo de peso do(s) classificador(es) votante(s) na nova categoria e (ii) no caso de novas consultas ao mesmo CNPJ, aumentando marginalmente o peso do(s) classificador(es) votante(s) como uma realimentação positiva. Observe neste último caso que foi dada uma interpretação de aceitação da classificação à repetição de uma consulta.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Até a data da elaboração deste documento, estas possibilidades encontram-se em fase de elaboração. Para fins desta meta, o ENSEMBLE atribui a mesma importância às crenças dos núcleos subordinados, ou seja, os pesos de cada um deles são iguais. Para cada subclasse (ou classe, grupo etc) é calculada a média imparcial das crenças informadas por cada um dos núcleos, e este novo conjunto de crenças compõe a resposta do ENSEMBLE. Este método de cálculo tem como características o reforço de crenças compartilhadas por vários núcleos e o enfraquecimento de crenças controversas.

O ENSEMBLE_CORE, conforme descrito acima, foi implementado no SCAE e seu desempenho está sendo avaliado.

3.6 Meta Física 2.1/2007 – Implementação de Protótipo do SCAE-Fiscal

O Sistema de Codificação Automática de Atividades Econômicas (SCAE) possui a arquitetura apresentada na Figura 3-19. O Sistema pode ser utilizado de duas formas, via linha de comando na máquina onde o SCAE está instalado, ou via navegador Internet (*browser*), que se comunica com o módulo Servidor de Aplicação (SA) do SCAE. Este, por sua vez, se comunica com os outros dois módulos do SCAE: Core e Banco de Dados (BD).

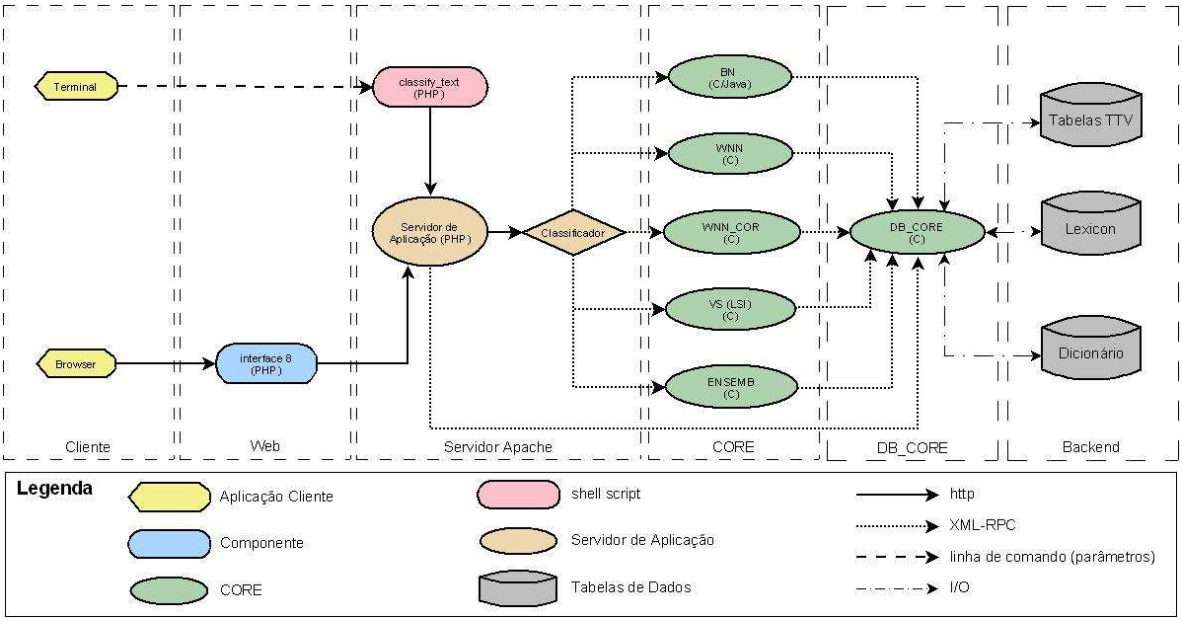

Figura 3-19: Arquitetura do SCAE

Em uma solicitação de classificação de atividade econômica, o usuário envia ao SA uma descrição de atividade econômica. O envio pode ser realizado das seguintes formas apresentadas abaixo:

- Pela linha de comando, na máquina onde está instalado o SCAE:

```
./classify_text <core> <objeto_a_classificar>

onde:

<core> - é o nome do core a ser utilizado: WNN, WNN_COR, VS, BN ou ENSEMB.
<objeto_a_classificar> - é um texto entre aspas que representa o objeto social
```

- Ou pelo browser, conforme apresentado na Figura 3-20:

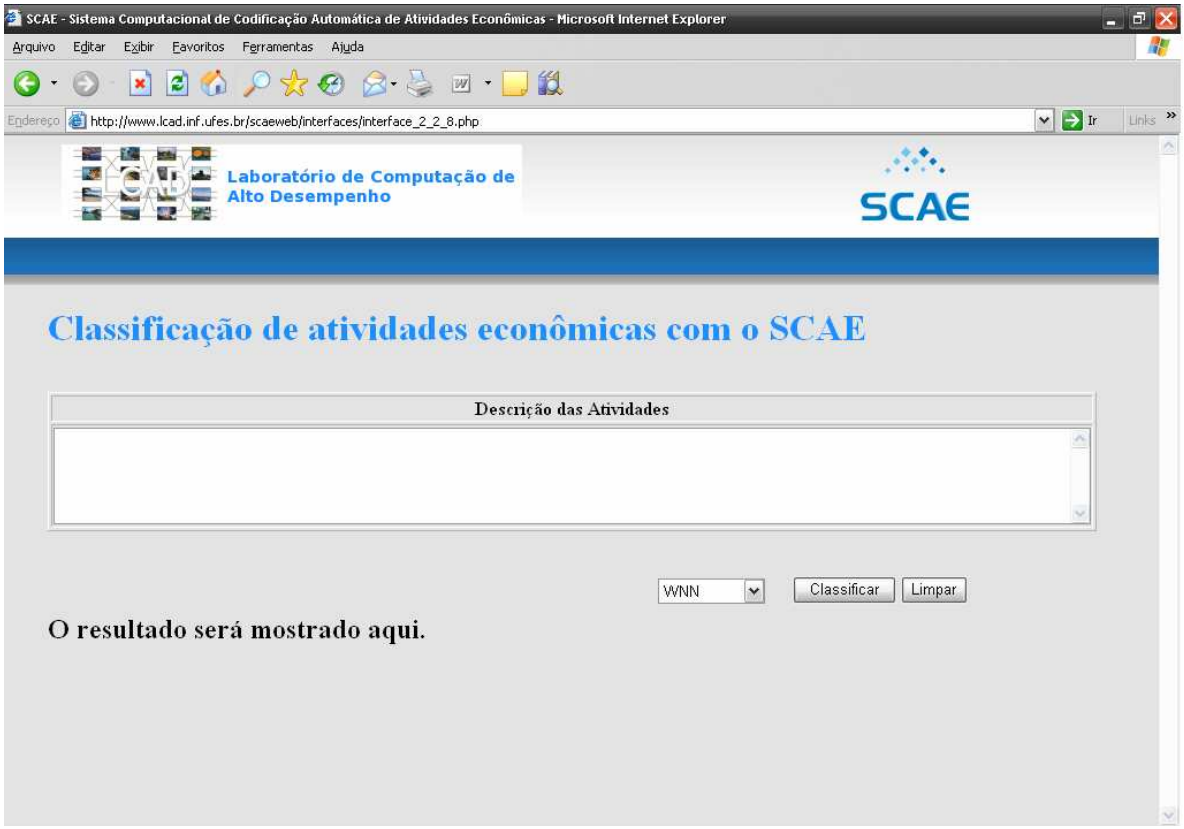

Figura 3-20: Interface WEB

O SA, por sua vez, envia esta descrição ao Core, que a classifica e retorna códigos CNAE e medidas de confiança quanto às associações destes códigos com a descrição de atividade econômica recebida. De posse dos códigos CNAE, o SA requisita ao BD seus textos associados.

O Core também se comunica com o módulo BD, que é responsável por armazenar todo o conhecimento do SCAE (dicionário eletrônico; representação, interna ao Sistema, da tabela CNAE e das descrições de atividades econômicas usadas em treino; etc.). Além de realizar a classificação, o Core manipula as tabelas do Sistema (insere dados, etc.) armazenadas no BD. Diferentes módulos do Core são responsáveis por esta manipulação. O SA, para esta Meta Física, foi desenvolvido em PHP, o BD em C e Java, e o Cores em C e Java.

3.6.1 Preparação para a Instalação

3.6.1.1 Requisitos Mínimos Necessários

Para instalar o SCAE em uma única máquina faz-se necessário que esta possua as seguintes características:

1. Espaço disponível: 30 GB ou superior.
2. Memória: 2 GB ou superior

3. Sistema Operacional: Fedora Core 6.

3.6.1.2 Preparando o Ambiente para Instalação do SCAE

A preparação do ambiente para instalação do SCAE consiste, basicamente, na instalação e configuração das bibliotecas de que este necessita. Abaixo são listados os pacotes e os passos necessários para sua instalação:

- w3c-libwww
- w3c-libwww-devel
- xmlrpc-c
- xmlrpc-c-devel
- xforms
- xforms-devel
- freeglut
- freeglut-devel
- libnet-devel
- byacc
- httpd
- php
- wine

Para instalação dos mesmos, efetue *login* como usuário *root* e entre com o comando a seguir para instalar os pacotes a partir do yum. Note que caso alguma delas já esteja instalada, o yum cuidará para que esta não seja instalada novamente e, caso alguma delas necessite de outra(s), o yum cuidará de instalar tais dependências. {Responda "y" quando for perguntado **"Is this ok [y/N]:"**.}

```
yum install -y w3c-libwww w3c-libwww-devel xmlrpc-c xmlrpc-c-devel xforms
xforms-devel freeglut freeglut-devel libnet-devel byacc httpd php wine
```

3.6.1.3 Aplicativos Instaláveis a partir do CD

JDK

O SCAE necessita de uma máquina virtual Java instalada na máquina onde estarão rodando o DB_CORE e o BN_CORE. A versão necessária é a 1.6.0_04. Para facilitar a instalação, foi disponibilizado no CD o arquivo binário correspondente a esta versão.

Para copiar o arquivo bin disponibilizado no CD para o /opt:

```
cp /media/SCAE/libs/jdk/jdk-6u4-linux-i586.bin /opt
```

Transforme este arquivo em um executável:

```
chmod +x /opt/jdk-6u4-linux-i586.bin
```

Instale o pacote com o comando a seguir:

```
cd /opt; ./jdk-6u4-linux-i586.bin ; ln -s jdk1.6.0_04 jdk-latest
```

A licença do pacote será exibida (aperte a tecla de espaço para passar cada tela) e, em seguida, você será perguntado se concorda com ela. Responda 'yes' para dar início à instalação. Em seguida, copie o script de configuração com os seguintes comandos, que também o tornam executável:

```
cp /media/SCAE/libs/jdk/java.sh /etc/profile.d/
chmod +x /etc/profile.d/java.sh
```

Agora execute o seguinte comando para tornar o arquivo disponível:

```
source /etc/profile.d/java.sh
```

Então, execute o seguinte comando para verificar se o *path* está correto:

```
which java
```

Você deverá obter algo como:

```
/opt/jdk-latest/bin/java
```

Agora execute os seguintes comandos:

```
/usr/sbin/alternatives --install /usr/bin/java java
/opt/jdk1.6.0_04/bin/java 2
/usr/sbin/alternatives --config java
```

Após ter inserido o último comando, você será perguntado sobre o tipo de Java que deseja para o sistema. Pressione 2 (a opção que indica o JDK 1.6.0_04) e pressione enter.

Para confirmar a versão do seu *Java Runtime Environment*, execute:

```
java –version
```

A saída esperada é:

```
java version "1.6.0_04"
Java(TM) SE Runtime Environment (build 1.6.0_04-b12)
Java HotSpot(TM) Client VM (build 10.0-b19, mixed mode, sharin
```

Por último, adicione as seguintes linhas ao seu arquivo .bashrc no seu diretório home:

```
#JAVA
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/jre/lib/i386
export LD_LIBRARY_PATH
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/jre/lib/i386/client
export LD_LIBRARY_PATH
```

MAE

Como root, copie a MAE para o /opt com o comando a seguir:

```
cp –r /media/SCAE/libs/MAE /opt/
```

Compile a MAE com o comando a seguir:

```
cd /opt/MAE; make clean; make -f Makefile.no_interface; cd
```

Insira as seguintes linhas ao seu .bashrc (localizado no seu diretório home):

```
# MAE
export MAEHOME=/opt/MAE
PATH=$PATH:$MAEHOME/bin
```

3.6.2 Instalando o SCAE

3.6.2.1 Instalando o Corretor Ortográfico

O SCAE utiliza como corretor ortográfico uma versão modificada do Corretor Ortográfico ASPELL. Com isto, existe apenas a necessidade de se configurar o diretório onde se encontram as bibliotecas desta versão.

Para configurar o corretor ortográfico do SCAE, inclua no arquivo .bashrc no seu *home* o caminho das bibliotecas do dicionário:

```
#SCAE SPELLER
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/relato3/CORES/DB_CORE/SPELLER/scaeaspell-
0.60.5/.libs
export LD_LIBRARY_PATH
```

Não se esqueça de atualizar as variáveis de ambiente em todos os terminais abertos com o comando:

```
source .bashrc
```

3.6.2.2 Instalando os Classificadores

Ainda como root, copie o SCAE para o diretório root com o comando:

```
cp -r /media/SCAE/relato3 /root
```

Em seguida, basta executar os comandos:

```
cd /root/relato3/CORES
./configure
make
make install
make test
```

O comando make compilará todos os CORES (ilustrados na Figura 3-19) existentes no SCAE.

Com a execução do comando make install, o SCAE adotará uma configuração default de treino e estará pronto a ser utilizado para fins de classificação. Contudo, caso se deseje adotar outras configurações de treino, vide a Seção 3.6.3. Nesta seção são apresentados mecanismos de construção de tabelas (Seção 3.6.3.1) e de execução de treino (Seção 3.6.3.5).

O comando make test assegurará que todos os CORES estão funcionando corretamente.

3.6.2.3 Instalando a Interface Web

Configure o servidor Apache para inicializar durante o *boot* da máquina:

```
chkconfig --level 5 httpd on
```

Caso o Apache não esteja ativo, inicialize-o com o comando:

```
/etc/init.d/httpd restart
```

Copie todo o conteúdo do diretório *relato3/scaeweb* para */var/www/html/*:

```
cp -r /root/relato3/scaeweb /var/www/html
```

Altere o usuário e o grupo do diretório scaweb para o usuário do Apache. Por padrão, o usuário e o grupo do servidor Apache 2.2.3, que já vem com o Fedora Core 6, é apache e apache, respectivamente:

```
chown apache:apache /var/www/html/scaeweb -R
```

Você pode então acessar o SCAE com um browser pela URL: http://127.0.0.1/scaeweb.

Seu browser deve mostrar a imagem da Figura 3-20 (pág. 51). Caso ela não apareça, há problemas de configuração no Apache ou PHP.

3.6.3 Configuração

3.6.3.1 Criação de Tabelas de Vetores de Treino e Teste (Tabelas TTV)

Após a instalação dos classificadores, podemos utilizar o **DB_CORE** para criar um conjunto básico de tabelas TTV que permitem demonstrar as funcionalidades do SCAE (assuma sempre que o diretório corrente é o último especificado por um comando cd). Para isto é necessário executar o script abaixo:

```
cd /root/relato3/CORES/DB_CORE; ./default_build.bat
```

A saída deste comando deve ser:

```
Locale set to pt_BR.UTF-8.
Loading known tables from .csv
Number of known tables = 29
Creating KNOWN_LEXICONS_saved.csv file.
Creating KNOWN_TTVS_saved.csv file.
Creating KNOWN_TRAININGS_saved.csv file.
```

```
Creating KNOWN_TESTS_saved.csv file.
Loading dictionary .csv
Number of dictionary words = 23160
Number of distincts words = 17365
Loading cnae subclasses .csv
Number of CNAE-Subclasses = 1183
Loading dados 'CSV_FILES/dados_vitoria_110.csv'
Number of economic activities descriptions = 14204
Number of distinct economic activities descriptions = 3281
Number of replicated cnae codes in the same activity = 232
Lexicon size = 1454. Number of words discarded due to word frequency
(PFS) = 0.
TTV 'TTV_C1S_DESC_TF' size = 1183
TTV 'TTV_DVS1_OBJS_TF' size = 3281
TTV 'TTV_C1S_DESC_TFIDF' size = 1183
TTV 'TTV_DVS1_OBJS_TFIDF' size = 3281
Saving tables...Done!
```

O script default_build.bat permite a criação das tabelas de treino e teste do SCAE. Ele realiza as seguintes ações, conforme demonstra a Figura 3-21:

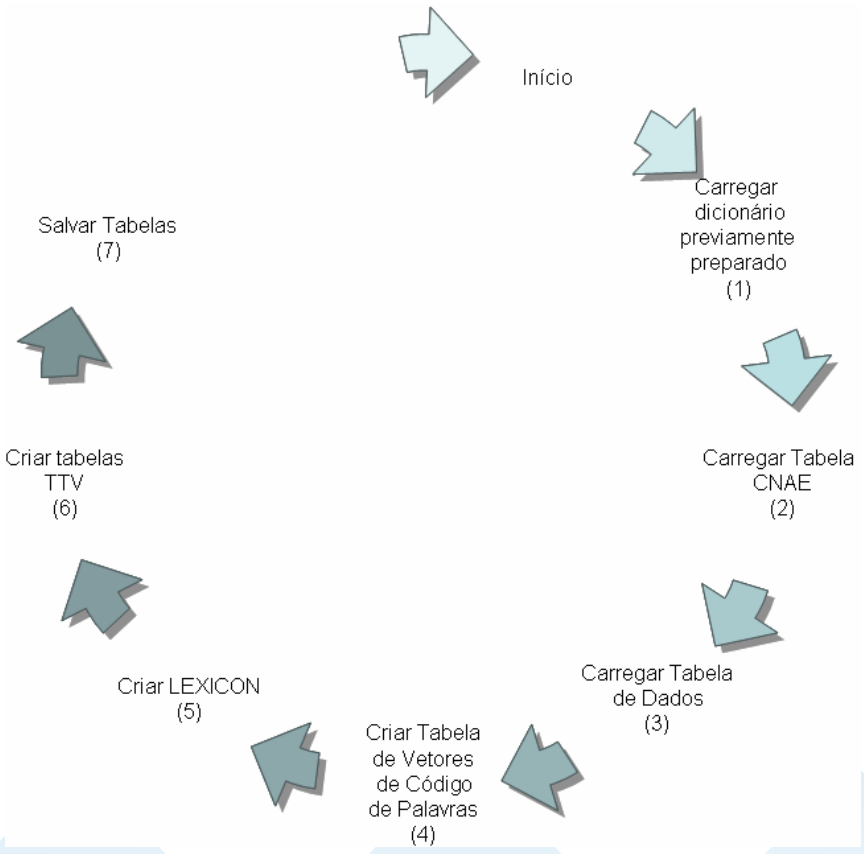

Figura 3-21: Processo de criação de tabelas do SCAE

As ações são executadas pelos seguintes comandos:

(1) `load_dictionary <NOME_DICIONARIO>`

(2) `load_csv_cnae_subclasse <NOME_TABELA_CNAE>`

(3) `load_csv_dados <NOME_TABELA_DADOS>`

(4) `create_word_vectors_table <<NOME_TABELA>-<NOME_CAMPO>>`

(5) `create_lexicon <NOME_LEXICON> <"<DESCRICAO_LEXICON>"> <NUMERO_TABELAS> <NOME_TABELA>-<<NOME_CAMPO>:<INI>:<FIM>> [<NOME_TABELA2>-<<NOME_CAMPO2>:<INI2>:<FIM2>>... <NOME_TABELAn>-<<NOME_CAMPOn>:<INIn>:<FIMn>>] <"[CLASSE_GRAMATICAL1] [CLASSE_GRAMATICAL2] ... [CLASSE_GRAMATICALn]"> <PFS>`

(6) `create_ttv <NOME_TTV> <NOME_LEXICON> <NOME_TABELA> <NUMERO_CAMPOS> <NOME_CAMPO> [<NOME_CAMPO2> ... <NOME_CAMPOn>] <TIPO_CONSTRUCAO> <PESO_TERMOS>`

(7) `save_tables_in_binary_format`

Os termos separados pelos sinais de maior e menor (<>) representam parâmetros obrigatórios e os termos separados por colchetes ([]) são opcionais. Tais parâmetros são explicados a seguir:

1. NOME_DICIONARIO – representa um dicionário preparado para o SCAE. Através de pesquisa exaustiva foram preparados vários dicionários. Estes dicionários foram gerados a partir de várias tabelas de dados, onde algumas destas tabelas foram fornecidas para o Projeto enquanto que outras foram obtidas por meio de pré-processamentos:
    a. NILC – Dicionário da Língua Portuguesa (Brasil), fornecido pelo Núcleo Interinstitucional de Lingüística Computacional (NILC)
    b. CNAE_110_SUBCLASSE – Tabela CNAE 1.1
    c. CNAE_110_SUBCLASSE_CORRIGIDO – Correção ortográfica manual da Tabela CNAE 1.1.
    d. DADOS_VITORIA_SUB – Tabela das descrições das atividades das empresas localizadas na região de Vitória (ES) com seus respectivos códigos CNAE.
    e. DADOS_VITORIA_SUB_CORRIGIDO – Correção ortográfica manual da tabela DADOS_VITORIA_SUB.
    f. DADOS_BH_SUB_110 – Tabela das descrições das atividades das empresas localizadas na região de Belo Horizonte (MG) com seus respectivos códigos CNAE.

Para a geração das bases foram utilizados os seguintes processos, algumas vezes em separado e outras em seqüência, conforme mostra a Tabela 3-1:

1. Correção Ortográfica Manual

2. Criação de Tabelas de Dicionário Canônico
3. Criação de Tabelas de Dicionário Radicalizado

Tabela 3-1: Lista de Tabelas de Dicionários do SCAE

Valor	Origem						Processo		
	A	B	C	D	E	F	1	2	3
DICIONARIO_SUBCLASSE		X						X	
DICIONARIO_COMPLETO	X							X	
DICIONARIO_COMPLETO_CORRIGIDO	X		X		X		X	X	
DICIONARIO_110_SUB+BH		X				X		X	
DICIONARIO_COMPLETO+BH	X					X		X	
DICIONARIO_COMPLETO+BH_SEM_N.C								X	
DICIONARIO_SEM_STOP_STEMM_SEM_ACCENT		X		X					X
DICIONARIO_SEM_STOP_STEMM		X		X					X

2. NOME_TABELA_CNAE – representa os tipos de Tabela CNAE existentes no DB_CORE. Atualmente existem os seguintes tipos:
    a. CNAE_110_SUBCLASSE – Tabela CNAE 1.1
    b. CNAE_110_SUBCLASSE_CORRIGIDO – Correção ortográfica manual (com intervenção humana) da Tabela CNAE 1.1.
    c. CNAE_110_SUBCLASSE_CORRIGIDO_AUTO – Correção ortográfica automática (sem intervenção humana, realizada com o corretor ortográfico do SCAE) da Tabela CNAE 1.1.
3. NOME_TABELA_DADOS – representa os tipos de Tabela de Dados existentes no DB_CORE. Suas tabelas foram geradas de forma análoga às Tabelas CNAE sendo que, tendo por base a tabela das descrições das atividades das empresas localizadas na região de Vitória (ES) com seus respectivos códigos CNAE. Atualmente existem os seguintes tipos:
    a. DADOS_VITORIA_SUB_110 – Tabela das descrições das atividades das empresas localizadas na região de Vitória (ES) com seus respectivos códigos CNAE-Subclasse da Tabela CNAE 1.1.
    b. DADOS_VITORIA_SUB_110_CORRIGIDO – Correção ortográfica manual (com intervenção humana) da tabela de dados DADOS_VITORIA_SUB_110.
    c. DADOS_VITORIA_SUB_110_CORRIGIDO_AUTO – Correção ortográfica automática (sem intervenção humana, realizada com o corretor ortográfico do SCAE) da tabela de dados DADOS_VITORIA_SUB_110.
4. NOME_TABELA – este parâmetro indica a tabela utilizada para gerar o vetor de palavras, o *lexicon* (conjunto de palavras que formam um vocabulário) ou as tabelas

de treino e teste. Atualmente, o sistema permite gerar o vetor de palavras para as seguintes tabelas:

a. Tabela CNAE – logo, o sistema reconhece os valores identificados no parâmetro 1 (NOME_TABELA_CNAE, acima). Caso seja desejado criar o vetor de palavras a partir desta tabela, o usuário deverá informar para o parâmetro NOME_CAMPO (que representa o campo de onde serão recuperados os descritores para se formar o vetor de palavras) o valor DESCRICAO_SUB.

b. Tabela de Dados – logo, o sistema reconhece os valores identificados no parâmetro 3 (NOME_TABELA_DADOS, acima). Caso seja desejado criar o vetor de palavras a partir desta tabela, o usuário deverá informar para o parâmetro NOME_CAMPO o valor OBJETO_SOCIAL.

5. NOME_LEXICON – este parâmetro indica o nome do *lexicon* a ser criado. Ele pode ser um texto livre, porém não deve conter espaços ou caracteres especiais. Contudo, sugere-se que o mesmo possua o seguinte padrão de nomenclatura:

   LEXICON[_LESS<[_ART][_CONJ][_CONTR][_INTERJ][_PREP][_PRON]>], onde os valores acima representam as classes gramaticais a serem removidas, conforme será explicado a seguir. Como exemplo, podemos citar: LEXICON, LEXICON_LESS_ART, LEXICON_LESS_CONJ, LEXICON_LESS_INTERJ, LEXICON_LESS_ART_PRON, etc.

6. CLASSE_GRAMATICAL – este parâmetro indica o nome da classe gramatical a ser excluída na construção do *lexicon*. Podem ser informados os seguintes valores para a classe gramatical:

   a. abr. – indica que removerá todos as abreviações.
   b. adj. – indica que removerá todos os adjetivos.
   c. adv. – indica que removerá todos os advérbio.
   d. art. – indica que removerá todos os artigos.
   e. conj. – indica que removerá todas as conjunções.
   f. contr. – indica que removerá todas as contrações
   g. interj. – indica que removerá todas as interjeições.
   h. nom. – indica que removerá todos os nomes próprios.
   i. num. – indica que removerá todos os numerais.
   j. prep. – indica que removerá todas as preposições.
   k. pron. – indica que removerá over todos os pronomes.
   l. sig. – indica que removerá todas as siglas.
   m. sub. – indica que removerá todos os substantivos.
   n. v. – indica que removerá todos os verbos.

A título de exemplo, a Tabela 3-2 indica um conjunto de nomes sugeridos de acordo com um subgrupo de classes gramaticais a serem removidas (apenas o subgrupo em que obtivemos os melhores resultados é mostrado):

Tabela 3-2: Exemplo de nomes para o *lexicon*

Nome sugerido	art.	conj.	contr.	Interj.	prep.	pron.
LEXICON						
LEXICON_LESS_ART	X					
LEXICON_LESS_CONJ		X				
LEXICON_LESS_ INTERJ				X		
LEXICON_LESS_ART_PRON	X					X

7. DESCRICAO_LEXICON – este parâmetro indica a descrição do *lexicon* a ser criado. Ele pode ser um texto livre. Sugere-se que o mesmo identifique as classes gramaticais que estão sendo removidas, conforme demonstrado na Tabela 3-3:

Tabela 3-3: Exemplo de descrições para o *lexicon*

Nome sugerido	art.	conj.	contr.	interj.	prep.	pron.
Lexicon sem remoção de classes gramaticais						
Lexicon sem artigo	X					
Lexicon sem conjunção		X				
Lexicon sem interjeição				X		
Lexicon sem artigo e sem pronome	X					X

8. NUMERO_TABELAS – este parâmetro indica o número de tabelas a serem utilizadas para a construção do *lexicon*.
9. INI – este parâmetro indica o número da linha inicial da tabela utilizada para a construção do *lexicon*.
10. FIM – este parâmetro indica o número da linha final da tabela utilizada para a construção do *lexicon*. Não pode ser informado um número maior do que o número de linhas da Tabela de Dados ou CNAE. Os limites existentes atualmente estão apresentados a seguir:

Tabela 3-4: Limites das Tabelas existentes atualmente no SCAE

Tabela	INI	FIM
DADOS_VITORIA_SUB_110	0	3280
DADOS_VITORIA_SUB_110_CORRIGIDO	0	3280
DADOS_VITORIA_SUB_110_CORRIGIDO_AUTO	0	3280
CNAE_110_SUBCLASSE	0	1182
CNAE_110_SUBCLASSE_CORRIGIDO	0	1182
CNAE_110_SUBCLASSE_CORRIGIDO_AUTO	0	1182

11. PFS – este parâmetro indica a freqüência acima da qual a palavra não será incluída no lexicon. Sugere-se informar um valor acima de 3000.
12. NOME_TTV – este parâmetro indica o nome da tabela de treino e teste. Ele pode ser um texto livre, porém não deve conter espaços ou caracteres especiais.
13. NUMERO_CAMPOS – este parâmetro indica o número de campos da tabela que será utilizada para a criação da TTV. Por default, utiliza-se apenas o número 1.
14. TIPO_CONSTRUCAO – este parâmetro indica o método de construção da TTV. Atualmente o DB_CORE aceita somente o valor DEFAULT.
15. PESO_TERMOS – este parâmetro denota a função para o cálculo dos pesos dos termos, que podem ser computados como a freqüência dos termos (*term frequency* (TF)) ou como a freqüência dos termos multiplicada pela freqüência inversa dos mesmos nos documentos (*term frequency inverse document frequency* (TFIDF)). Atualmente este parâmetro permite os valores TF e TFDIF.

3.6.3.2 Criação de Tabelas de Dicionários

Criação de Tabelas de Dicionários Canônicos

No diretório Diadorim, localizado dentro do DB_CORE, existem scripts (em *perl* e *shell*) que são empregados para gerar um dicionário com palavras canônicas a partir de tabelas. Os passos estão esboçados abaixo:

(1) Extração de palavras a partir tabelas: execute, para tanto, o script abaixo:

```
perl extrai_palavras.pl <NOME_TABELA> <SAIDA1>
```

(2) Remoção de duplicatas da lista de palavras extraídas. Utilize o script abaixo:

```
perl remove_duplicatas_da_lista_palavras.pl <SAIDA1> <SAIDA2>
```

(3) Execução do aplicativo Diadorim (também desenvolvido pelo NILC) sobre a lista palavras (já sem repetição). Assegure, antes, que o *wine* esteja instalado.

```
wine CLSView.exe MSGR2PB.lex <SAIDA2> <SAIDA3>
```

(4) Remover entradas duplicatas do dicionário gerado:

```
perl remove_duplicatas_do_dicionario.pl <SAIDA3> <SAIDSA4>
```

(5) Converter a saída do Diadorim para o formato que o BD interpreta (.csv). Ao final da execução dos comandos abaixo, um arquivo de nome dicionário.csv será gerado no mesmo diretório:

```
mv SAIDA4 dicionario.txt
perl remove_duplicatas_do_dicionario.pl
```

Criação de Tabelas de Dicionários Radicalizados

Após a instalação dos classificadores, é possível também criar dicionários radicalizados utilizando o procedimento de *Stemming*. Para isto, é necessário executar o script `build_filtered_dictionary_from_tables.bat`, que usa o executável **db_core** para criar um dicionário filtrado. O script é apresentado abaixo.

```
./db_core
create_filtered_dictionary_from_tables
<NOME_DICIONARIO_A_GERAR> <ID_FILTRO> <"[[OPERACAO1] [+
<OPERACAO2> [+ <OPERACAO3>]]"> <NUMERO_TABELAS>
<NOME_TABELA>-<NOME_CAMPO> [<NOME_TABELA2>-<NOME_CAMPO2> ..
<NOME_TABELAn>-<NOME_CAMPOn>]
```

onde:

1. NOME_DICIONARIO_A_GERAR – representa o nome do dicionário a ser construído para o SCAE.
2. ID_FILTRO – um número inteiro identificando o filtro. Atualmente o sistema admite apenas um único valor (1);
3. OPERACAO – este parâmetro indica as operações que podem ser realizadas pelo filtro. Ele permite informar os seguintes valores:

Tabela 3-5: Operações de Filtro do SCAE

Operação	Descrição
STEMM	Permite reduzir as palavras existentes no dicionário na forma radicalizada
ACCENT	Permite remover os acentos das palavras
STOP	Permite remover as palavras que são consideradas *stop words* do dicionário.

As operações podem ser combinadas em até oito possibilidades, incluindo a opção de não se informar a operação. Neste caso deve-se informar branco (“”).

3.6.3.3 Correção Ortográfica das Bases

Após a instalação dos classificadores, é possível realizar a correção ortográfica das Bases utilizando o corretor ortográfico do SCAE.

Primeiramente, um dicionário baseado no formato do *Aspell* e uma lista de palavras com respectivas freqüências precisam ser gerados. Por padrão, o SCAE já possui o dicionário e a lista gerados, não havendo necessidade, por parte do usuário, de gerá-los.

O dicionário, arquivo dict_scae.rws, está localizado no diretório DB_CORE/SPELLER/dictionary e a lista de palavras, words_frequency.csv, está localizada no diretório DB_CORE/CSV_FILES.

A correção ortográfica de uma Base é realizada pelo *script* build_corrected_tables.bat localizado no diretório DB_CORE. O *script* é apresentado abaixo:

```
(1) db_core
(2) check_table            <NOME_LISTA_PALAVRA_FREQUENCIA>
    <NOME_TABELA1> <NUMERO_CAMPOS1> <NOME_CAMPO1>
    [check_table <NOME_LISTA_PALAVRA_FREQUENCIA>
    <NOME_TABELA2> <NUMERO_CAMPOS2> <NOME_CAMPO2> \
                          ....
    check_table <NOME_LISTA_PALAVRA_FREQUENCIA>
    <NOME_TABELAn> <NUMERO_CAMPOSn> <NOME_CAMPOn> \]
```

Os termos separados pelos sinais de maior e menor (<>) representam parâmetros obrigatórios. Tais parâmetros são explicados a seguir:

1. NOME_LISTA_PALAVRA_FREQUENCIA – representa a lista de palavras com as respectivas freqüências. Atualmente, o SCAE permite informar o valor WORDS_FREQUENCY.
2. NOME_TABELA – representa o nome da tabela a ser corrigida. Os possíveis valores para este parâmetro são os apresentados em NOME_TABELA_CNAE e NOME_TABELA_DADOS;

3. NUMERO_CAMPOS – este parâmetro indica o número de campos da tabela que será utilizada para a correção. Por default, utiliza-se apenas o número 1;
4. NOME_CAMPO – este parâmetro indica o nome do campo da tabela a ser corrigido. Os possíveis valores para este parâmetro foram explicados anteriormente.

Para executar o *script* digite na linha de comando:

```
cd /root/relato3/CORES/DB_CORE; ./build_corrected_tables.bat
```

Após a correção ortográfica, a tabela corrigida é salva no diretório DB_CORE/CSV_FILES com o mesmo nome da tabela original, mas com o prefixo _corrigido_auto.

3.6.3.4 Criação das Novas Bases

A fim de gerar as quatro bases propostas na Seção 3.7.1 (pág. 76), um novo script foi inserido no diretório do DB_CORE. Ao executar o script generate_base.bat, contanto que ele possua os parâmetros adequados, é possível construir as bases nas condições especificadas na na Seção 3.7.1. O script é apresentado abaixo.

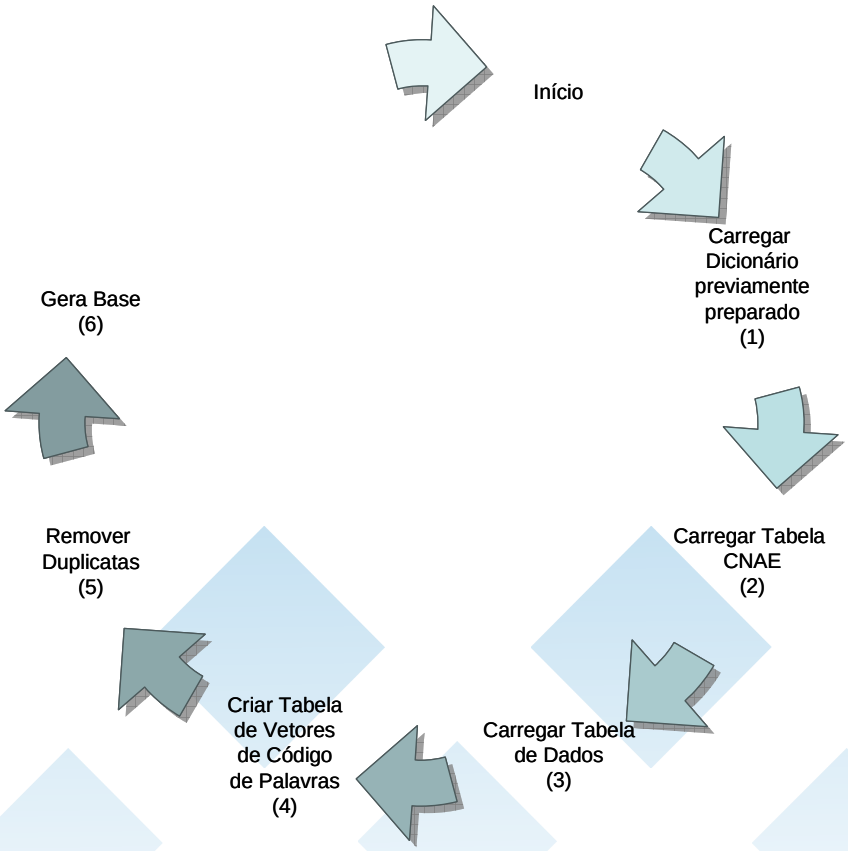

Figura 3-22: Processo de criação de novas bases

As ações (1), (2), (3) e (4) foram descritas anteriormente, e as ações (5) e (6) são executadas pelos seguintes comandos:

```
(5)    remove_duplicate <BASE_SEM_DUPLICATAS>
  <NUMERO_TABELAS_DE_DADOS> <NOME_TABELA>-<NOME_CAMPO>
  [<NOME_TABELA2>-<NOME_CAMPO2>... <NOME_TABELAn>-
  <NOME_CAMPOn>] <NUMERO_TABELAS_CNAE> <NOME_TABELA>-
  <NOME_CAMPO> [<NOME_TABELA2>-<NOME_CAMPO2>...
  <NOME_TABELAn>-<NOME_CAMPOn>]

(6)    generate_base <NOME_BASE_GERADA> <FUNÇÃO>
  <LIMITE> <NUMERO_BASES_SEM_DUPLICATAS> <
  BASE_SEM_DUPLICATAS> [<BASE_SEM_DUPLICATAS2>...
  <BASE_SEM_DUPLICATASn>]
```

Novamente, os termos separados pelos sinais de maior e menor (<>) representam parâmetros obrigatórios. Muitos destes parâmetros já foram apresentados anteriormente. Apresentam-se, agora, os demais:

1. BASE_SEM_DUPLICATAS – representa o nome da tabela, sem entradas duplicadas, a ser gerada a partir de duas ou mais tabelas de entradas.
2. NUMERO_TABELAS_DE_DADOS – este parâmetro indica o número de tabelas de DADOS que serão utilizadas na construção da base sem duplicatas.
3. NUMERO_TABELAS_CNAE – este parâmetro indica o número de tabelas CNAE, no caso, que serão utilizadas na construção da base sem duplicatas.
4. NOME_BASE_GERADA – este parâmetro indica o nome da base que será gerada (os nomes foram propostos na Seção 3.7.1, pág. 76).
5. FUNÇÃO – este parâmetro assume dois valores: PISO e TETO. O primeiro indica que não serão incluídos códigos com freqüência inferior a um dado limite (apresentando abaixo). Caso o parâmetro seja TETO, todos os documentos relacionados aos códigos com freqüência inferior ao limite serão incluídos na nova base.
6. LIMITE – este parâmetro indica a freqüência que, dependendo do tipo de função (apresentada acima) determinará se o código será ou não incluído na nova base.
7. NUMERO_BASES_SEM_DUPLICATAS – este parâmetro indica o número de bases sem duplicatas que serão utilizadas para criar uma nova base seguindo as especificações apresentadas na Seção 3.7.1.

A tabela abaixo apresenta os parâmetros adotados para a criação das bases propostas na Seção 3.7.1 utilizando o script apresentado acima. Ainda na Tabela 3-6, o nome DADOS_BH+VIT está relacionado à base, sem duplicatas, gerada a partir das bases de Vitória e Belo Horizonte.

Tabela 3-6: Parâmetros para criação das novas bases.

NOME_BASE	FUNÇÃO	LIMITE	ENTRADA
BASEVBH1	PISO	100	DADOS_BH+VIT
BASEVBH2	TETO	100	DADOS_BH+VIT
BASEVBH3	PISO	30	DADOS_BH+VIT
BASEVBH4	TETO	30	DADOS_BH+VIT

3.6.3.5 Treino

Após criarmos as tabelas de treino e teste no DB_CORE, podemos realizar o treino dos CORES categorizadores. O pré-requisito para realizar o treino de um CORE é que esse e o DB_CORE estejam ativos, ou seja, "escutando" numa determinada porta.

O treino é realizado por meio de *scripts* localizados no diretório relato3/CORES/USER_INTERFACE/. A Tabela 3-7 mostra relação entre os scripts e os CORES.

Tabela 3-7: Scripts de treino dos CORES

Nome do script	Treino do CORE
wnn_default_train.bat	WNN
wnn_cor_default_train.bat	WNN_COR
vs_default_train.bat	VS
bn_default_train.bat	BN
ensemb_default_train.bat	ENSEMB

Para realizar o treino de um determinado CORE, execute o seguinte comando:

```
./<nome do script>
```

Caso tenha escolhido realizar o treino do WNN_COR, o script apresenta a seguinte mensagem na tela do terminal:

```
Locale set to pt_BR.UTF-8.

SERVERS'S STATUS:
*** DB is ON ***
*** WNN is OFF ***
*** VS is OFF ***
*** WNN_COR is ON ***
*** BN is OFF ***
*** ENSEMB is OFF ***
Training WNN_COR CORE ...
Training successfully finished.
```

```
Saving TREINAMENTO_DEFAULT in WNN_COR CORE was completed.
```

O script de treino realiza as seguintes ações:

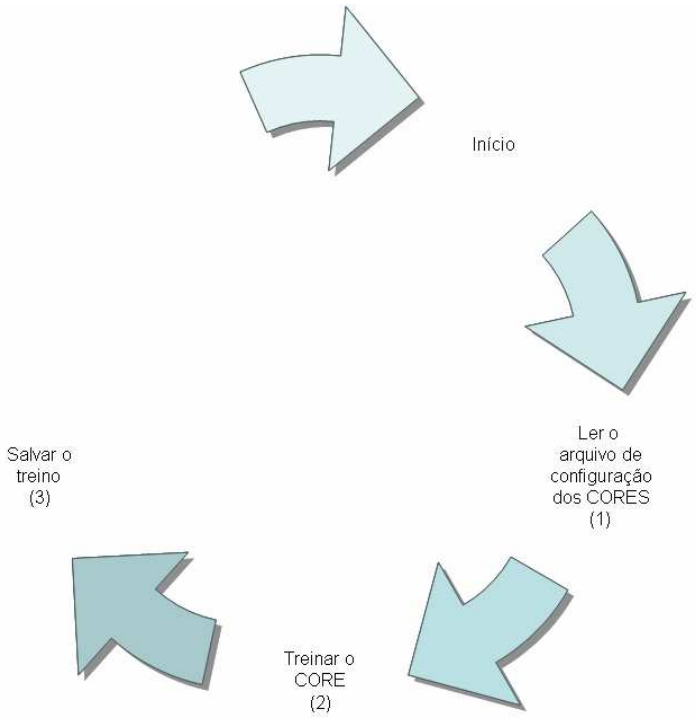

Figura 3-23: Ações realizadas no treino do CORE

As ações são executadas pelos seguintes parâmetros:

(1) `read_ports ports.cfg`

(2) `train <NOME_CORE> <NOME_TREINO> <"<DESCRICAO_TREINO>"> <NUMERO_TTVS> <NOME_TTV1 INI1 FIM1> [<NOME_TTV2 INI2 FIM2> ... <NOME_TTVn INIn FIMn>]`

(3) `save <NOME_CORE> <"<NOME_TREINO>">`

Os termos separados pelos sinais de maior e menor (<>) representam parâmetros obrigatórios. Tais parâmetros são explicados a seguir:

1. NOME_CORE – representa o nome do CORE que será treinado. Atualmente, os seguintes nomes são aceitos: WNN, WNN_COR, VS, BN e ENSEMB;
2. NOME_TREINO – representa o nome do treino. Pode ser um texto livre, porém não deve conter espaços ou caracteres especiais;

3. DESCRICAO_TREINO – é uma descrição do treino a ser realizado. Ele pode ser um texto livre. Sugere-se que o mesmo identifique as tabelas de treino e teste que estão sendo utilizadas. A descrição do treino deve ser colocado entre aspas duplas(“);
4. NUMERO_TTVS – indica o número de tabelas de treino e teste que serão utilizadas para o treino;
5. NOME_TTV – indica o nome da tabela de treino e teste criada na execução do `default_build.bat`;
6. INI – número da linha inicial da tabela de treino e teste para o treino do CORE;
7. FIM – número da linha final da tabela de treino e teste para o treino do CORE. Não pode ser informado um número maior do que o número de linhas da Tabela de Dados ou CNAE. Os limites existentes foram apresentados na Tabela 3-4.

Os CORES, com exceção do ENSEMB, salvam o treinamento em um diretório localizado no diretório de cada CORE. A Tabela 3-8 mostra a relação entre o nome do CORE, o diretório do mesmo e o diretório de treino.

Tabela 3-8: Relação entre o nome do CORE e os diretórios de treino.

Nome do CORE	Diretório do CORE	Diretório de treino
WNN	WNN_CORE	WNN_MEMORIES
WNN_COR	WNN_COR_CORE	WNN_MEMORIES
VS	VS_CORE	VECTORS_SPACES
BN	BN_CORE	NETWORKS
ENSEMB	ENSEMB_CORE	-

3.6.4 Uso

Atualmente, o SCAE disponibiliza duas funcionalidades de uso: o teste e a classificação de atividades econômicas. Os procedimentos para utilizá-las são apresentados nas seções 3.6.4.2 e 3.6.4.3.

3.6.4.1 Gerenciamento dos CORES

Além de permitir ao usuário a realização de procedimentos de treino e teste, o SCAE disponibiliza um script para gerenciamento dos CORES. Este script pode ser encontrado em /etc/init.d.

Através deste script é possível:

- Inicializar todos os CORES:

```
/etc/init.d/classifier_cores start
```

- Verificar status de todos os CORES:

```
/etc/init.d/classifier_cores status
```

- Parar todos os CORES:

```
/etc/init.d/classifier_cores stop
```

3.6.4.2 Testes

Após treinarmos um determinado CORE, podemos realizar o teste do desempenho do mesmo segundo diversas métricas (ver Seção 3.7.3, pag. 115, para a descrição das métricas). O pré-requisito para realizar o teste de um CORE é que esse e o DB_CORE estejam ativos, ou seja, "escutando" numa determinada porta, e o que CORE tenha sido treinado.

O teste é realizado por meio de *scripts* localizados no diretório relato3/CORES/USER_INTERFACE/. A Tabela 3-9 mostra a relação entre os *scripts* e o nome dos CORES.

Tabela 3-9: Scripts de teste dos CORES

Nome do script	Teste do CORE
wnn_default_test.bat	WNN
wnn_cor_default_test.bat	WNN_COR
vs_default_test.bat	VS
bn_default_test.bat	BN
ensemb_default_test.bat	ENSEMB

Para realizar o teste de um determinado CORE, execute o seguinte comando:

```
./<nome do script>
```

Caso tenha escolhido realizar o teste do WNN_COR, o script apresenta a seguinte mensagem na tela do terminal:

```
Locale set to pt_BR.UTF-8.

SERVERS'S STATUS:
*** DB is ON ***
*** WNN is OFF ***
*** VS is OFF ***
*** WNN_COR is ON ***
*** BN is OFF ***
*** ENSEMB is OFF ***
Reloading training TREINAMENTO_DEFAULT in WNN_COR CORE ...
Training TREINAMENTO_DEFAULT in WNN_COR CORE was reloaded successfully.
Testing WNN_COR CORE ...
```

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

```
#### METRICS ####
Ordinal ranking - One Error = 0,010969
Ordinal ranking - Ranking Loss = 0,000115
Ordinal ranking - Coverage = 3,424132
Ordinal ranking - Average Precision^d = 0,990275
Ordinal ranking - R-Precision^d = 0,981901
Ordinal ranking - Hamming Loss = 0,000061
Ordinal ranking - R-Hamming Loss = 0,036198
Ordinal ranking - Microaveraged Precision = 0,991600
Ordinal ranking - Microaveraged Recall = 0,991600
Ordinal ranking - Macroaveraged Precision^d = 0,981901
Ordinal ranking - Macroaveraged Recall^d = 0,981901
Ordinal ranking - Macroaveraged Precision^c = 0,579381
Ordinal ranking - Macroaveraged Recall^c = 0,577633
Ordinal ranking - Microaveraged F_1 = 0,991600
Ordinal ranking - Macroaveraged F_1^d = 0,981901
Ordinal ranking - Macroaveraged F_1^c = 0,578506
Modified competition ranking - One Error = 0,012797
Modified competition ranking - Ranking Loss = 0,000115
Modified competition ranking - Coverage = 3,444241
Modified competition ranking - Average Precision^d = 0,682200
Modified competition ranking - R-Precision^d = 0,981810
Modified competition ranking - Hamming Loss = 0,000063
Modified competition ranking - R-Hamming Loss = 0,035430
Modified competition ranking - Microaveraged Precision = 0,993281
Modified competition ranking - Microaveraged Recall = 0,989180
Modified competition ranking - Macroaveraged Precision^d = 0,981810
Modified competition ranking - Macroaveraged Recall^d = 0,979297
Modified competition ranking - Macroaveraged Precision^c = 0,580157
Modified competition ranking - Macroaveraged Recall^c = 0,576296
Modified competition ranking - Microaveraged F_1 = 0,991226
Modified competition ranking - Macroaveraged F_1^d = 0,980552
Modified competition ranking - Macroaveraged F_1^c = 0,578220
```

As seguintes ações são realizadas pelo script de teste:

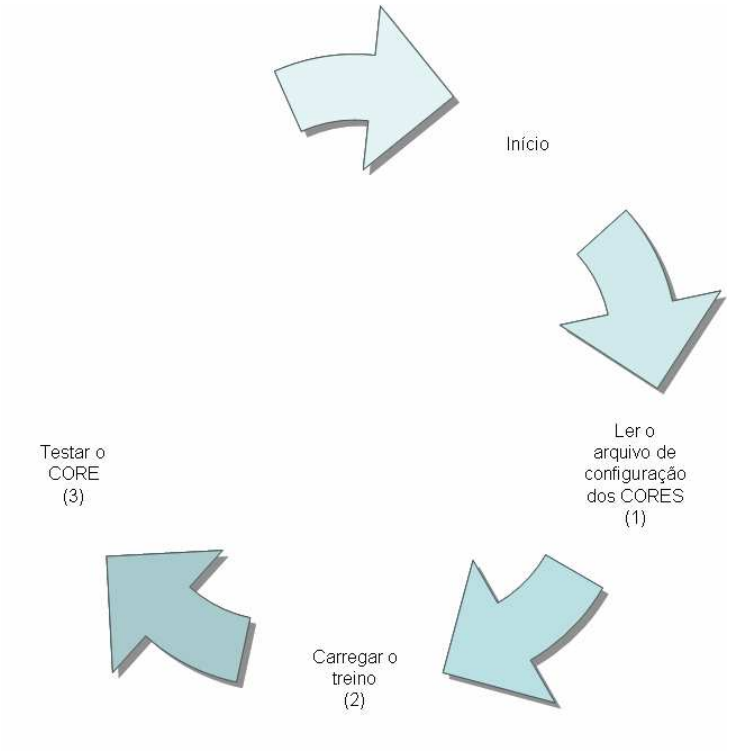

Figura 3-24: Ações realizadas no teste do CORE

As ações são executadas pelos seguintes parâmetros:

(1) `read_ports ports.cfg`

(2) `reload <NOME_CORE> <NOME_TREINO>`

(3) `test <NOME_CORE> <NIVEL_CNAE> <NOME_TREINO> <"<DESCRICAO_TESTE>"> <NUMERO_TTVS> <NOME_TTV1 INI1 FIM1> [<NOME_TTV2 INI2 FIM2> ... <NOME_TTVn INIn FIMn>]`

Os termos separados pelos sinais de maior e menor (<>) representam parâmetros obrigatórios. Tais parâmetros são explicados a seguir:

1. NOME_CORE – representa o nome do CORE que será treinado. Atualmente, os seguintes nomes são aceitos: WNN, WNN_COR, VS, BN e ENSEMB;
2. NOME_TREINO – representa o nome do treino utilizado para treinar o CORE;
3. NIVIL_CNAE – indica em qual nível CNAE será realizado o teste. Os níveis CNAE podem ser: SECAO, DIVISAO; GRUPO, CLASSE e SUBCLASSE;
4. DESCRICAO_TESTE – é uma descrição do teste a ser realizado. Ele pode ser um texto livre. Sugere-se que o mesmo identifique as tabelas de treino e teste que estão sendo utilizadas. A descrição do teste deve ser colocada entre aspas duplas (“”);
5. NUMERO_TTVS – indica o número de tabelas de treino e teste que serão utilizadas para o teste;

6. NOME_TTV – indica o nome da tabela de treino e teste criada na execução do `default_build.bat`;
7. INI – número da linha inicial da tabela de treino e teste para o teste do CORE;
8. FIM – número da linha final da tabela de treino e teste para o teste do CORE. Não pode ser informado um número maior do que o número de linhas da Tabela de Dados ou CNAE. Os limites existentes foram apresentados na Tabela 3-4.

3.6.4.3 Classificação de Atividades Econômicas

Além do treino e teste, o sistema SCAE permite que descrições de atividades, na forma de texto livre, sejam classificadas (categorizadas), ou seja, dada uma descrição de atividade, o sistema retorna possíveis códigos CNAE-Subclasse. A classificação de atividades econômicas pode ser feita de duas maneiras pelo sistema: uma é utilizar o *browser*, e a outra é utilizar o script `classify_text.bat` localizado no diretório relato3/CORES/USER_INTEFACE.

Classificação via Browser

Na primeira opção, com o *browser* aberto, o usuário deve digitar a URL http://127.0.0.1/scaeweb. Como conseqüência, deve aparecer uma página *Web* semelhante à Figura 3-25. Para realizar a classificação, o usuário precisa: digitar a descrição da atividade econômica no campo Descrição das Atividades, e selecionar qual CORE que será utilizado para realizar a classificação. Atualmente, as opções de CORE disponíveis no SCAE são WNN, WNN_COR, VS(LSI), BN e ESEMB.

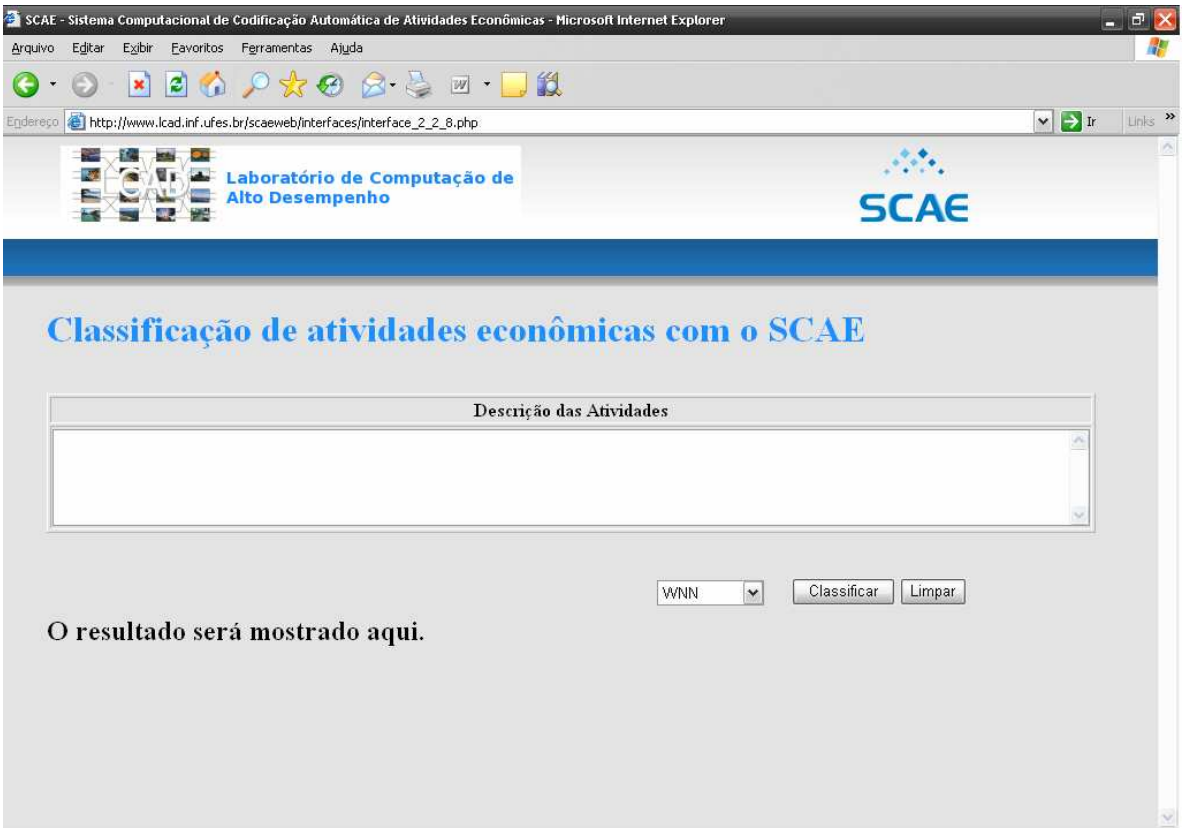

Figura 3-25: Interface Web de classificação de atividades.

Após preencher todas as informações necessárias, o usuário deve clicar no botão Classificar para classificar a atividade descrita. Ao clicar no botão Classificar, uma mensagem "Classificando" é mostrada informando o status da classificação da atividade. Para apagar as informações digitadas na página, o usuário pode clicar no botão Limpar ou pressionar a tecla F5.

O resultado da classificação é mostrado no item "O resultado será mostrado aqui.". Esse item, também, é utilizado para informar os possíveis erros decorrentes de comunicação com os *CORES*, mensagens de erro XML, campos preenchidos incorretamente, etc. A Figura 3-26 mostra a classificação da atividade Cultivo de arroz utilizando o WNN_COR.

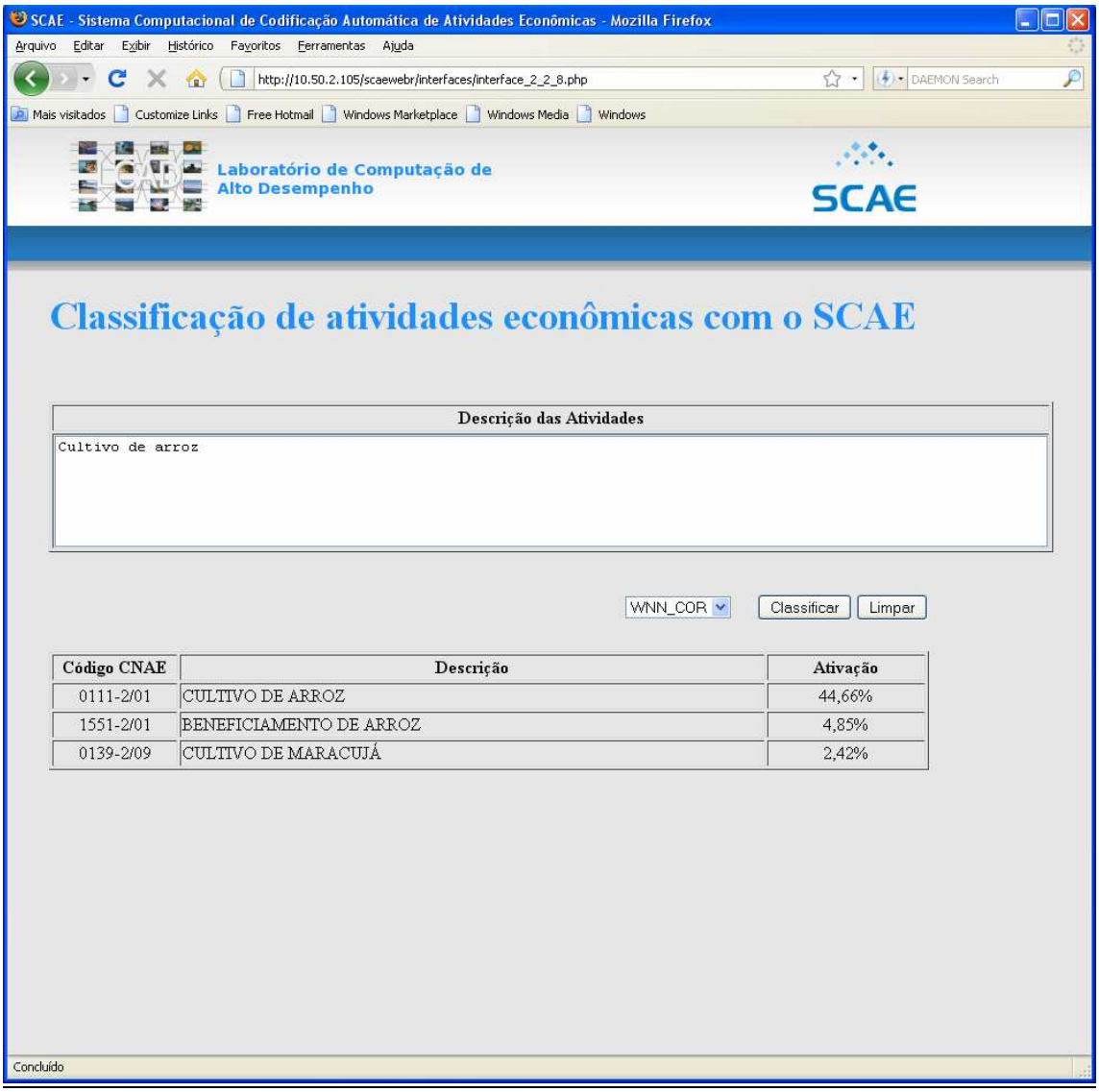

Figura 3-26: Classificação da atividade Cultivo de arroz com o CORE WNN_COR pelo *browser*

O resultado da classificação é mostrado no formato de tabela, conforme Figura 3-26, onde a primeira coluna representa o código CNAE-Subclasse, a segunda a descrição do código (denominação do código segundo a tabela CNAE) e a terceira a Ativação, no caso do WNN e WNN_COR, ou Cosseno do Ângulo, no caso do VS(LSI), ou Crença, no caso do BN e ENSEMB, atribuída pelo CORE.

Classificação via Script

Para realizar a classificação de atividades econômicas utilizando o *script*, o usuário deve editar o arquivo `classify_text.bat` localizado no diretório relato3/CORES/USER_INTEFACE e executar o seguinte comando:

```
./classify_text.bat
```

A ação de classificação é executada pelo seguinte parâmetro:

(1) `classify_text <"<NOME_CORE>"> <"<DESCRICAO ATIVIDADE>">`

Os termos separados pelos sinais de maior e menor (<>) representam parâmetros obrigatórios. Tais parâmetros são explicados a seguir:

1. NOME_CORE – representa o nome do CORE que será treinado. Atualmente, os seguintes nomes são aceitos: WNN, WNN_COR, VS, BN e ENSEMB. O nome do CORE deve ser colocado entre aspas duplas (“”);
2. DESCRICAO_ATIVIDADE – representa a descrição da atividade a ser classificada. É um texto livre que deve colocado entre aspas duplas (“”).

Por padrão, esse *script* é configurado para classificar a descrição de atividade Cultivo de arroz utilizando o WNN_COR:

```
./user_interface \
classify_text "WNN_COR" "Cultivo de arroz."
```

A mensagem retornada pelo script é a seguinte:

```
Locale set to pt_BR.UTF-8.
CORE_NAME  : "WNN_COR"
TEXT TYPED : "Cultivo de arroz."
0111-2/01       CULTIVO DE ARROZ         44,66%
1551-2/01       BENEFICIAMENTO DE ARROZ 4,85%
0139-2/09       CULTIVO DE MARACUJÃ      2,42%
```

A segunda linha indica o nome do CORE selecionado, a terceira linha a descrição da atividade econômica para classificação e as linhas restantes representam a classificação da atividade pelo CORE, sendo que a primeira coluna representa o código CNAE-Subclasse, a segunda o descritor e a terceira a Ativação (ou Crença, ou Cosseno do ângulo) atribuída pelo CORE.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

3.7 Meta Física 3.1/2007 – Criação de *Benchmarking* para Realização de Comparações entre os Métodos

No Projeto *Classificação Automática em CNAE-Fiscal*, no item metodologia, ficou definido que a disponibilização das seguintes bases de dados, em formato eletrônico de fácil importação para banco de dados (o mais apropriado para o Projeto seria XML), seria feita por parte da Receita Federal:

- A Tabela CNAE-Fiscal completa (com as Notas Explicativas e Descritores); e
- Uma base de dados representativa de objetos sociais e suas codificações manuais segundo a Tabela CNAE-Fiscal.

A base de dados representativa de codificações manuais consiste de uma amostra representativa, do ponto de vista estatístico, de objetos sociais e suas codificações em CNAE-Fiscal. Esta base de dados será utilizada para a calibração dos subsistemas matematico-computacionais de codificação que estão sendo desenvolvidos.

Nós definimos o número significativo mínimo necessário de objetos sociais utilizando os conceitos de variabilidade estatística, de forma a garantir que a amostra contenha todas as nuances que tragam variabilidade para a tarefa de classificação (exemplos: códigos com diferentes graus de dificuldade de classificação – fáceis, médios e difíceis; diferentes incidências de códigos de acordo com a região; diversidade de códigos; etc...). Estas fontes de variação podem resultar em estratificações importantes para a coleta de dados e serão definidas juntamente com os técnicos que hoje fazem classificação.

Para validar a versão final do SCAE-Fiscal, uma outra amostra representativa com um número significativo de objetos sociais deverá ser fornecida pela Receita Federal até o fim do projeto.

Na Seção 3.7.1, a seguir, é apresentada metodologia para se obter bases de dados representativas de codificações manuais, propõe-se um sistema on-line de avaliação automática à distância para classificadores de atividades econômicas, e também uma análise descritiva do banco de dados de descrições de atividades econômicas e códigos CNAE associados de Vitória/ES e Belo Horizonte/MG. Na Seção 3.7.2 são apresentadas novas bases de dados obtidas a partir da base de dados de Vitória/ES e Belo Horizonte/MG, e na Seção 3.7.3 são discutidas métricas para avaliação dos métodos de categorização de texto em desenvolvimento.

3.7.1 Definição das Bases de Dados Representativas

As bases de dados representativas a serem definidas serão utilizadas para:

a) aferir a capacidade de classificação dos classificadores manuais;
b) estudar os modelos quanto à sua capacidade de resolver o problema proposto no projeto, considerando as particularidades;
c) calibrar os modelos automatizados propostos (baseados em aprendizado de máquina);
d) aferir a capacidade de classificação dos modelos propostos;

e) comparar estatisticamente estes modelos.

Para atender aos múltiplos objetivos das bases de dados, propomos que elas sejam montadas com várias fontes de informação. A Tabela 3-10, abaixo, apresenta a proposta, com as bases, os objetivos, as vantagens, dificuldades e soluções.

Tabela 3-10: Bases de dados representativas

BASE (DESCRIÇÃO)	OBJETIVO	VANTAGENS	DIFICULDADES/SOLUÇÕES
BASE 1: dados da central de dúvidas do IBGE (texto da atividade principal, perguntas do IBGE e o código atribuído pelos especialistas).	Esta base será utilizada para a aferição e calibração dos classificadores manuais. Pode ser utilizada posteriormente para avaliação dos modelos propostos.	Os dados desta base podem ser agrupados como fáceis, medianos e difíceis e, além disso, possuem o resultado da codificação dada pelos especialistas do IBGE.	Os dados da base de dados estão armazenados em e-mails e deverão ser retirados manualmente (dispomos de equipe para obtê-los). Além disso, vamos precisar dos especialistas do IBGE para agrupar as atividades nos graus fácil, mediano e difícil (contamos com a parceria do IBGE).
BASE 2: dados de objetos sociais das prefeituras de Vitória e Belo Horizonte (texto das atividades e códigos atribuídos pelos órgãos)	Estudar os modelos quanto a sua capacidade de resolver o problema proposto no projeto, considerando as particularidades. Não será utilizada para aprendizagem dos modelos e nem para validá-los.	São bases já disponibilizadas e de tamanhos e variabilidades que permitem o entendimento do problema.	Não foi aferida a confiabilidade dos códigos atribuídos às atividades. Base limitada para validação dos modelos.
BASE 3: Dados da pesquisa econômica do IBGE. Em torno de 30 mil empresas. Dados da atividade principal, perguntas e códigos.	Calibrar os modelos automatizados propostos e aferir a capacidade de classificação dos mesmos.	Esta base será obtida em entrevista por telefone, dando a liberdade ao entrevistador de fazer perguntas complementares. Dados atuais.	Como o objetivo da pesquisa do IBGE é outro, precisamos conversar com os coordenadores para ajustar alguns detalhes importantes para a nossa pesquisa (usar o campo outras observações para registrar perguntas adicionais, por exemplo).
BASE 4: Base montada para os experimentos do projeto. Será criado um protótipo de entrada de dados com texto livre de atividades e perguntas adicionais, e estes dados serão classificados pelos classificadores manuais	Calibrar os modelos automatizados propostos e aferir a capacidade de classificação dos mesmos. Será ainda útil para comparar estatisticamente os modelos.	As características desta base vão responder às principais questões inerentes ao projeto.	Por ser uma base grande (estima-se um número de documentos da ordem de 60 mil), será necessário um grande esforço para obtenção dos dados e para a codificação manual. Nossa equipe dará todo o suporte para a motivação do trabalho e obtenção dos dados.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Base 1

A Base 1 é composta por casos tratados pela central de dúvidas do IBGE (texto da atividade principal, perguntas do IBGE e o código atribuído pelos especialistas). Solicitamos um total de 360 casos, distribuídos segundo o grau de dificuldade (fácil, médio e difícil) e também a grande área CNAE (serviço, indústria, comércio e agricultura). A Tabela 3-11, abaixo, apresenta os totais solicitados. Caso uma das grandes áreas tenha dificuldade de apresentar o número de casos (30 por grau de dificuldade), podemos limitar em 20 casos.

Tabela 3-11: Descrição da Base 1

Dificuldade	Área				TOTAL
	serviço	indústria	comércio	agricultura	
Fácil	30	30	30	30	120
Médio	30	30	30	30	120
Difícil	30	30	30	30	120
TOTAL	90	90	90	90	360

OBS.: Caso seja difícil respeitar esta estratificação podemos re-calcular os tamanhos das amostras.

Foi indicado ao IBGE o formato para a disponibilização desta base: Os dados deverão vir em arquivos com o número do caso, grau de dificuldade e grandes áreas da tabela CNAE. Abaixo apresentamos o modelo solicitado:

CASO: <colocar o número>

GRAU DE DIFICULDADE: <completar com fácil, médio ou difícil>

ÁREA: <completar com serviço, indústria, comércio ou agricultura>

- e-mail origem:

<Pergunta do contribuinte>

- e-mail pergunta IBGE 1:

<pergunta 1 adicional do IBGE>

- e-mail resposta do contribuinte pergunta 1:

<resposta do contribuinte à pergunta 1 do IBGE>

- **.....**

- e-mail pergunta IBGE n:

<pergunta n adicional do IBGE>

- e-mail resposta do contribuinte pergunta n:

<resposta do contribuinte à pergunta n do IBGE>

- e-mail resposta final do IBGE

<possíveis códigos>

Apresentamos abaixo um exemplo de dados já recebidos:

Caso 23
Grau de Dificuldade: Médio
Área: Atividades de atenção à saúde

- e-mail origem:

Estou registrando uma médica especialista em cirurgia plástica. Preciso saber qual o nº de CNAE que devo usar?

- e-mail pergunta IBGE 1:

Prezado usuário,

A empresa possui consultório próprio?

- e-mail resposta do contribuinte pergunta 1:

Não é uma empresa, é uma profissional liberal que presta serviços dentro de uma clinica, mas não tem vinculo com a mesma.

- e-mail pergunta IBGE n:
(NE)
- e-mail resposta do contribuinte pergunta n:
(NE)
- e-mail resposta final do IBGE

8630-5/99 ATIVIDADES DE ATENÇÃO AMBULATORIAL NÃO ESPECIFICADAS ANTERIORMENTE

Base 2

A Base 2 é composta por objetos sociais de empresas de Vitória e Belo Horizonte e seus códigos CNAE associados (texto das atividades e códigos atribuídos pelas prefeituras de Vitória e Belo Horizonte).

Dados de Vitória – ES

A seguir são apresentadas algumas estatísticas descritivas dos dados de Vitória – ES.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Tabela 3-12: Dados Vitória - Número de códigos por documento

Números de Códigos por Documento	Freqüência	%
1	837	25,51
2	617	18,81
3	494	15,06
4	348	10,61
5	234	7,13
6	180	5,49
7	162	4,94
8	92	2,80
9	61	1,86
10	51	1,55
11	22	0,67
12	22	0,67
13	16	0,49
14	22	0,67
15	14	0,43
16	12	0,37
17	18	0,55
18	9	0,27
19	10	0,30
20	5	0,15
21	3	0,09
22	5	0,15
23	5	0,15
24	5	0,15
25	3	0,09
26	4	0,12
27	2	0,06
28	3	0,09
29	3	0,09
30	2	0,06
31	1	0,03
32	1	0,03
33	2	0,06
34	1	0,03
35	1	0,03
42	1	0,03
43	1	0,03
44	4	0,12
46	1	0,03
51	1	0,03
63	1	0,03
64	1	0,03
67	1	0,03
68	1	0,03
99	1	0,03
109	1	0,03
Total	3281	100,00

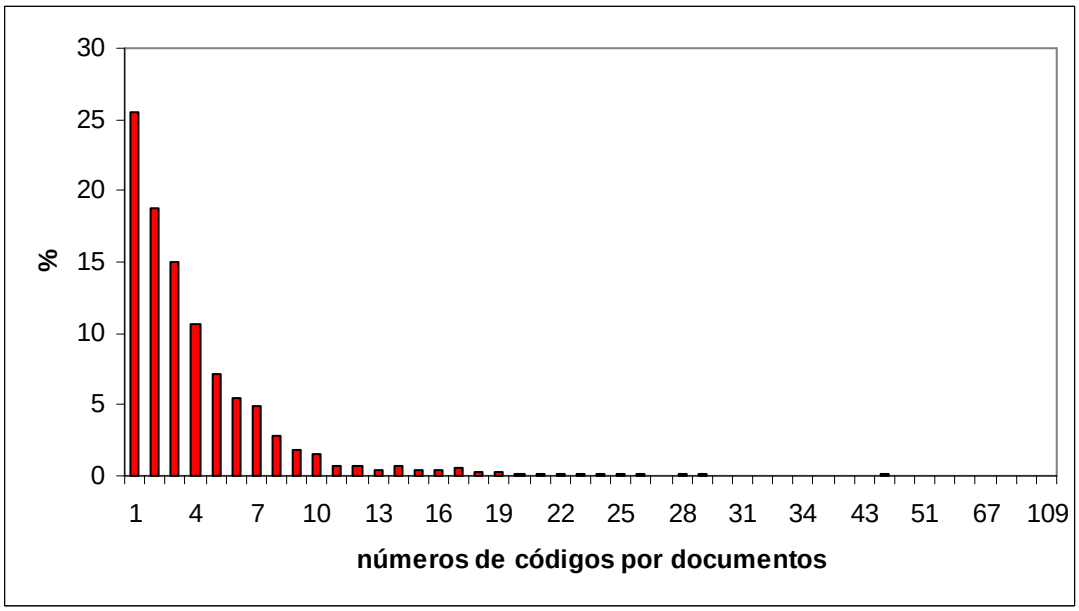

Figura 3-27: Dados Vitória - Número de códigos por documento

Tabela 3-13: Dados Vitória - Estatísticas descritivas para o número de códigos por documento

Mínimo	Máximo	Média	Mediana	Moda	Desvio Padrão
1	109	4,3	3	1	5,6

Verifica-se que, a maior parte, 25,51% num total de 3281 documentos, recebeu apenas um código, o valor máximo de códigos por documento foi de 109 e o número médio de códigos por documento foi aproximadamente igual a 4, neste banco de dados de Vitória. Verifica-se também, através do desvio padrão, que a variabilidade dos dados é alta.

Tabela 3-14: Dados Vitória - Distribuição de freqüências por Seção

Seção	Freqüência	%
A	61	0,43
B	1	0,01
C	74	0,52
D	560	3,91
E	26	0,18
F	733	5,12
G	6943	48,51
H	400	2,79
I	646	4,51
J	655	4,58
K	3100	21,66
L	11	0,08
M	347	2,42
N	235	1,64
O	521	3,64
Total	14313	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

A Seção G (comercio; reparação de veículos automotores, objetos pessoais e domésticos, conforme a Tabela CNAE-FISCAL 1.1) foi a mais freqüente, superando 48% num total de 14313 códigos atribuídos.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Tabela 3-15: Dados Vitória - Distribuição de freqüências por Divisão

Divisão	Freqüência	%
1	56	0,39
2	5	0,03
5	1	0,01
10	1	0,01
11	13	0,09
13	10	0,07
14	50	0,35
15	27	0,19
17	5	0,03
18	62	0,43
19	15	0,1
20	3	0,02
21	2	0,01
22	155	1,08
23	3	0,02
24	14	0,1
25	6	0,04
26	31	0,22
27	6	0,04
28	11	0,08
29	83	0,58
30	8	0,06
31	17	0,12
32	9	0,06
33	35	0,24
34	3	0,02
35	10	0,07
36	53	0,37
37	2	0,01
40	13	0,09
41	13	0,09
45	733	5,12
50	456	3,19
51	2594	18,12
52	3893	27,2
55	400	2,79
60	164	1,15
61	13	0,09
62	9	0,06
63	327	2,28
64	133	0,93
65	265	1,85
66	91	0,64
67	299	2,09
70	257	1,8
71	242	1,69
72	678	4,74
73	38	0,27
74	1885	13,17
75	11	0,08
80	347	2,42
85	235	1,64
90	26	0,18
91	121	0,85
92	208	1,45
93	166	1,16
Total	14313	100

A maior parte dos códigos, aproximadamente 27%, pertencia à Divisão 52 (Comércio varejista e reparação de objetos pessoais e domésticos) e aproximadamente 18% pertenciam à

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Divisão 51 (Comércio por atacado e representantes comerciais e agentes do comércio). As divisões 51 e 52 pertencem à Seção G (Comércio; reparação de veículos automotores, objetos pessoais e domésticos, conforme a Tabela CNAE-FISCAL 1.1).

Tabela 3-16: Dados Vitória - Distribuição de freqüências por Grupo

Grupo	Freqüência	%
524	2308	16,13
523	990	6,92
514	754	5,27
749	664	4,64
741	588	4,11
511	550	3,84
552	384	2,68
515	350	2,45
452	333	2,33
516	312	2,18
513	310	2,17
809	299	2,09
742	285	1,99
522	274	1,91
722	235	1,64
519	220	1,54
851	211	1,47
521	174	1,22
671	174	1,22
503	171	1,19
634	166	1,16
930	166	1,16
454	161	1,12
602	158	1,10
659	158	1,10
744	140	0,98
725	137	0,96
713	136	0,95
527	131	0,92
703	129	0,90
672	125	0,87
745	116	0,81
455	112	0,78
501	110	0,77
723	109	0,76
502	107	0,75
512	98	0,68
222	89	0,62
919	87	0,61
642	84	0,59
923	83	0,58
721	79	0,55
729	79	0,55
926	75	0,52
299	72	0,50
451	72	0,50
Total	14313	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Cento e sessenta e sete diferentes Grupos foram observados. O Grupo mais freqüente neste banco de dados foi o 524 (Comércio varejista de outros produtos), estando presente em 16,13%, num total de 14313 códigos. A Tabela 3-16 apresenta somente os Grupos que apresentaram freqüência superior a 0,5%.

Tabela 3-17: Dados Vitória - Distribuição de freqüência por Classe

Classe	Freqüência	%
52493	809	5,65
74993	605	4,23
52337	390	2,72
74160	359	2,51
52329	352	2,46
51195	340	2,38
52450	312	2,18
80993	291	2,03
74209	285	1,99
52426	273	1,91
52434	268	1,87
52469	259	1,81
52310	248	1,73
52418	214	1,50
51497	199	1,39
51535	188	1,31
55212	184	1,29
51918	175	1,22
50300	171	1,19
52442	168	1,17
63401	166	1,16
55220	164	1,15
67199	161	1,12
45217	157	1,10
51691	148	1,03
72290	146	1,02
65994	143	1,00
51390	142	0,99
74403	140	0,98
72508	137	0,96
51454	133	0,93
67202	125	0,87
93025	118	0,82
74500	116	0,81
45500	112	0,78
50105	110	0,77
72303	109	0,76
50202	107	0,75
51446	104	0,73
51659	104	0,73
85138	103	0,72
51365	98	0,68
51462	98	0,68
52159	97	0,68
52299	92	0,64
72214	89	0,62
60267	85	0,59
64203	84	0,59
74128	82	0,57
72109	79	0,55
72907	79	0,55
51420	78	0,54
52710	77	0,54
45411	76	0,53
70327	75	0,52
51217	72	0,50
Total	14313	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Foram observados 364 diferentes classes. A classe mais freqüente neste banco de dados foi a 52493 (Comércio varejista de outros produtos não especificados anteriormente), estando presente em 5,65%, num total de 14313 códigos. A Tabela 3-17 apresenta somente as classes que apresentam uma porcentagem superior a 0,5%.

Tabela 3-18: Dados Vitória - Distribuição de freqüências por Subclasse

Códigos	Freqüência	%
7416002	353	2,47
5232900	352	2,46
5119500	340	2,38
5249303	235	1,64
5233701	209	1,46
5245002	193	1,35
5233702	181	1,26
7499399	178	1,24
7420902	167	1,17
5522000	164	1,15
5241804	148	1,03
5191801	146	1,02
7229000	146	1,02
5242601	142	0,99
5246902	142	0,99
7250800	137	0,96
8099305	136	0,95
4521701	124	0,87
5231002	124	0,87
5243499	118	0,82
6719999	116	0,81
5249306	111	0,78
6720201	111	0,78
6599403	109	0,76
7230300	109	0,76
7499312	108	0,75
5249399	93	0,65
5521201	92	0,64
5521202	92	0,64
7499307	91	0,64
5215902	90	0,63
7221400	89	0,62
7440301	89	0,62
5231003	88	0,61
5249302	85	0,59
5245003	79	0,55
7210900	79	0,55
7290700	79	0,55
5243401	75	0,52
7032700	75	0,52
5229999	72	0,50
7450002	71	0,50
8099399	71	0,50
Total	14313	100,00

Foram observadas 764 diferentes Subclasses (Códigos). Cento e cinqüenta e oito dessas subclasses (20,68%) estavam presentes em apenas um documento. De acordo com a tabela acima (Tabela 3-18), pode-se observar que as Subclasses mais freqüentes neste banco de dados foram a 7416002 (Atividades de assessoria em gestão empresarial) e 5232900 (Comércio varejista de artigos do vestuário e complementos), com percentual de 2,47% e 2,46% respectivamente, num total de 14313 códigos. A Tabela 3-18 apresenta somente as Subclasses que apresentaram freqüência superior a 0,5%, num total de 14313 códigos.

Dados de Belo Horizonte - MG

A seguir são apresentadas algumas estatísticas descritivas dos dados de Belo Horizonte - MG.

Tabela 3-19: Dados BH - Número de códigos por documento

Números de Códigos por Documento	Freqüência	%
1	48930	55,602
2	18349	20,851
3	9572	10,877
4	4775	5,426
5	2699	3,067
6	1415	1,608
7	801	0,910
8	555	0,631
9	297	0,338
10	210	0,239
11	110	0,125
12	80	0,091
13	39	0,044
14	41	0,047
15	28	0,032
16	21	0,024
17	13	0,015
18	11	0,013
19	33	0,038
20	5	0,006
21	3	0,003
22	6	0,007
23	2	0,002
24	2	0,002
25	1	0,001
26	1	0,001
27	1	0,001
Total	88000	100,000

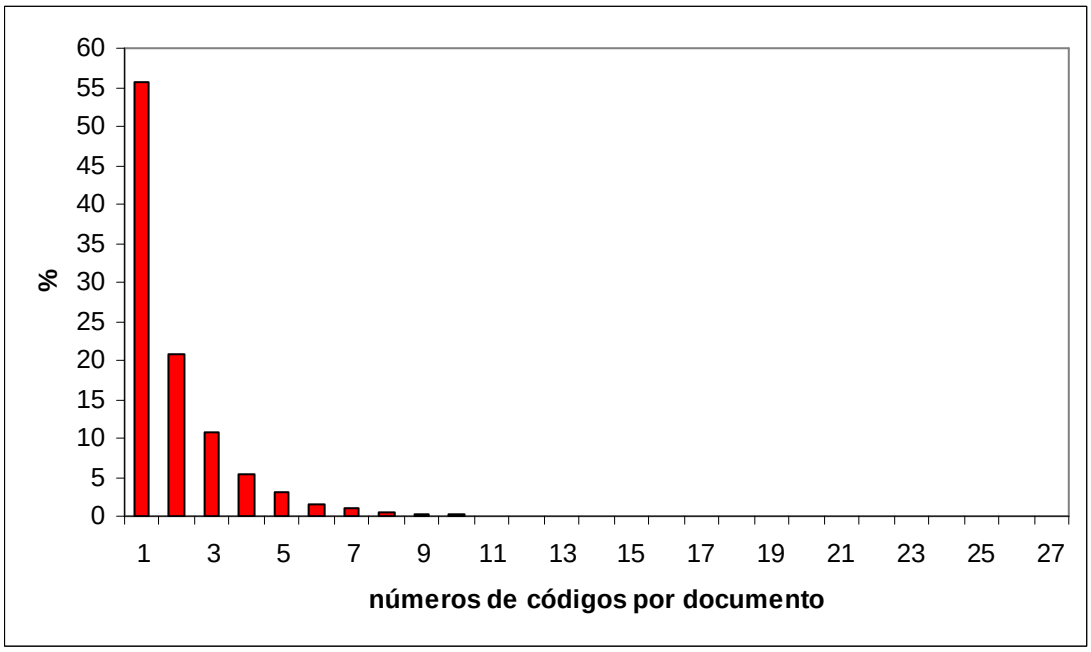

Figura 3-28: Dados BH - Número de códigos por documento

Tabela 3-20: Dados BH - Estatísticas descritivas para o número de códigos por documento

Mínimo	Máximo	Média	Mediana	Moda	Desvio padrão
1	27	2,0	1	1	1,7

Verifica-se que, a maior parte, aproximadamente 55,6% num total de 88000 documentos, recebeu apenas um código, o valor máximo de códigos por documento foi de 27 e o número médio de códigos por documento foi igual a 2.

Tabela 3-21: Dados BH - Distribuição de freqüências por Seção

Seção	Freqüência	%
A	429	0,24
B	9	0,01
C	161	0,09
D	11360	6,46
E	152	0,09
F	9400	5,35
G	68500	38,98
H	6645	3,78
I	6242	3,55
J	5668	3,23
K	49223	28,01
L	233	0,13
M	5527	3,14
N	4223	2,40
O	7966	4,53
P	13	0,01
Total	175751	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

A Seção G (Comércio; reparação de veículos automotores, objetos pessoais e domésticos, conforme a tabela CNAE 1.1) foi a mais freqüente, superando 38% num total de 175751 códigos atribuídos.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Tabela 3-22: Dados BH - Distribuição de freqüências por Divisão

Divisão	Freqüência	%
01	381	0,217
02	48	0,027
05	9	0,005
10	4	0,002
13	82	0,047
14	75	0,043
15	861	0,490
17	232	0,132
18	1234	0,702
19	174	0,099
20	137	0,078
21	73	0,042
22	3345	1,903
23	5	0,003
24	221	0,126
25	137	0,078
26	220	0,125
27	76	0,043
28	855	0,486
29	1854	1,055
30	27	0,015
31	277	0,158
32	109	0,062
33	539	0,307
34	129	0,073
35	61	0,035
36	764	0,435
37	30	0,017
40	138	0,079
41	14	0,008
45	9400	5,348
50	8592	4,889
51	15309	8,711
52	44599	25,376
55	6645	3,781
60	2941	1,673
61	10	0,006
62	72	0,041
63	1985	1,129
64	1234	0,702
65	2976	1,693
66	336	0,191
67	2356	1,341
70	12713	7,234
71	3519	2,002
72	7949	4,523
73	325	0,185
74	24717	14,064
75	233	0,133
80	5527	3,145
85	4223	2,403
90	263	0,150
91	2309	1,314
92	3563	2,027
93	1831	1,042
95	13	0,007
Total	175751	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Cinqüenta e seis diferentes Divisões foram observadas. A maior parte dos códigos, aproximadamente 25%, pertencia à Divisão 52 (Comércio; reparação de veículos automotores, objetos pessoais e domésticos, de acordo com a tabela CNAE 1.1) e aproximadamente 14% pertenciam à Divisão 74 (Atividades imobiliárias, aluguéis e serviços prestados as empresas).

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Tabela 3-23: Dados BH - Distribuição de freqüências por Grupo

Grupo	Freqüência	%
524	25677	14,610
523	8635	4,913
704	8595	4,890
741	8046	4,578
749	6586	3,747
511	6373	3,626
552	6107	3,475
742	5734	3,263
522	5482	3,119
452	4726	2,689
851	3922	2,232
514	3917	2,229
809	3823	2,175
502	3344	1,903
503	2965	1,687
602	2933	1,669
722	2510	1,428
527	2404	1,368
521	2254	1,282
222	2155	1,226
744	2071	1,178
713	1911	1,087
919	1895	1,078
454	1878	1,069
672	1843	1,049
930	1831	1,042
659	1766	1,005
516	1610	0,916
701	1587	0,903
703	1582	0,900
299	1562	0,889
501	1525	0,868
725	1476	0,840
455	1453	0,827
729	1421	0,809
926	1393	0,793
515	1354	0,770
801	1333	0,758
923	1233	0,702
513	1214	0,691
181	1181	0,672
221	1129	0,642
723	1097	0,624
745	1095	0,623
721	1069	0,608
702	949	0,540
Total	175751	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Duzentos e nove diferentes Grupos foram observados. Os Grupos mais freqüentes neste banco de dados foram o 524 (Comércio varejista de outros produtos, de acordo com a tabela CNAE 1.1) e o 523 (Comércio varejista de tecidos, artigos de armarinho, vestuário e calçados), com percentual de aproximadamente 14% e 5% respectivamente, num total de 175751 códigos.

A tabela acima (Tabela 3-23) apresenta somente os Grupos que apresentam uma porcentagem superior a 0,5%.

Tabela 3-24: Dados BH - Distribuição de freqüências por Classe

Classe	Freqüência	%
70408	8595	4,890
52493	8436	4,800
74993	5866	3,338
74209	5734	3,263
74160	5311	3,022
52329	4280	2,435
80993	3746	2,131
52442	3628	2,064
50202	3344	1,903
45217	3227	1,836
52418	3048	1,734
50300	2965	1,687
51195	2921	1,662
55212	2896	1,648
52450	2810	1,599
52469	2765	1,573
55220	2715	1,545
52310	2565	1,459
52434	2439	1,388
52426	2230	1,269
74403	2071	1,178
51187	2056	1,170
85138	2017	1,148
52299	1945	1,107
67202	1843	1,049
60267	1816	1,033
52337	1790	1,018
52132	1721	0,979
22292	1631	0,928
65994	1604	0,913
70106	1587	0,903
50105	1525	0,868
72508	1476	0,840
45500	1453	0,827
72907	1421	0,809
91995	1383	0,787
74128	1347	0,766
52213	1292	0,735
72214	1267	0,721
72290	1243	0,707
52795	1211	0,689
93025	1169	0,665
52710	1156	0,658
45411	1129	0,642
52248	1098	0,625
72303	1097	0,624
74500	1095	0,623
72109	1069	0,608
51497	1050	0,597
51691	1005	0,572
70327	990	0,563
71390	979	0,557
92312	965	0,549
85154	956	0,544
29963	951	0,541
70203	949	0,540
18120	900	0,512
Total	175751	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Foram observados 519 diferentes Classes. A Classe mais freqüente foi a 70408 (Condomínios prediais, de acordo com a tabela CNAE 1.1) com percentual de aproximadamente 5%, num total de 175751 códigos. A tabela acima (Tabela 3-24) apresenta somente as classes que apresentam uma porcentagem superior a 0,5%.

Tabela 3-25: Dados BH - Distribuição de freqüências por Subclasse

Subclasse	Freqüência	%
7040800	8595	4,890
7416002	5247	2,985
5232900	4280	2,435
7420902	4004	2,278
5119500	2921	1,662
5249399	2863	1,629
4521701	2830	1,610
5522000	2715	1,545
8099305	2136	1,215
5118700	2056	1,170
5020201	1995	1,135
5030003	1945	1,107
5249303	1891	1,076
5245002	1791	1,019
5521201	1718	0,978
5231002	1647	0,937
5246902	1643	0,935
7010600	1587	0,903
5213202	1577	0,897
7250800	1476	0,840
5242601	1434	0,816
7290700	1421	0,809
9199500	1383	0,787
7499399	1356	0,772
5229999	1312	0,747
7221400	1267	0,721
5241804	1259	0,716
7229000	1243	0,707
8513801	1196	0,681
5233701	1193	0,679
7440301	1185	0,674
5521202	1178	0,670
5243401	1111	0,632
5224800	1098	0,625
7230300	1097	0,624
6720201	1086	0,618
6599403	1077	0,613
7210900	1069	0,608
4541101	1044	0,594
7412801	1003	0,571
7499302	993	0,565
7032700	990	0,563
7499307	978	0,556
5244201	952	0,542
7020300	949	0,540
2996399	911	0,518
7499308	889	0,506
6026701	881	0,501
Total	175751	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Foram observados 1002 diferentes Subclasses (códigos). Oitenta dessas Subclasses (7,98%) estavam presentes em apenas um documento. De acordo com a tabela acima (Tabela 3-25), pode-se observar que a Classe mais freqüente foi a 7040800 (Condomínios de prédios residenciais ou não, de acordo com a tabela CNAE 1.1), com percentual de aproximadamente 5%. A tabela acima (Tabela 3-25) apresenta somente as Subclasses que apresentaram freqüência superior a 0,5%, num total de 175751 códigos.

Dados das Cidades de Belo Horizonte – MG e Vitória – ES Combinados

A seguir são apresentadas algumas estatísticas descritivas dos dados de Vitória – ES e Belo Horizonte combinados.

Tabela 3-26: Dados de Vitória e BH combinados - Número de códigos por documento

Números de Códigos por Documentos	Freqüência	%
1	49767	54,521
2	18966	20,778
3	10066	11,027
4	5123	5,612
5	2933	3,213
6	1595	1,747
7	963	1,055
8	647	0,709
9	358	0,392
10	261	0,286
11	132	0,145
12	102	0,112
13	55	0,060
14	63	0,069
15	42	0,046
16	33	0,036
17	31	0,034
18	20	0,022
19	43	0,047
20	10	0,011
21	6	0,007
22	11	0,012
23	7	0,008
24	7	0,008
25	4	0,004
26	5	0,005
27	3	0,003
28	3	0,003
29	3	0,003
30	2	0,002
31	1	0,001
32	1	0,001
33	2	0,002
34	1	0,001
35	1	0,001
42	1	0,001
43	1	0,001
44	4	0,004
46	1	0,001
51	1	0,001
63	1	0,001
64	1	0,001
67	1	0,001
68	1	0,001
99	1	0,001
109	1	0,001
Total	91281	100,00

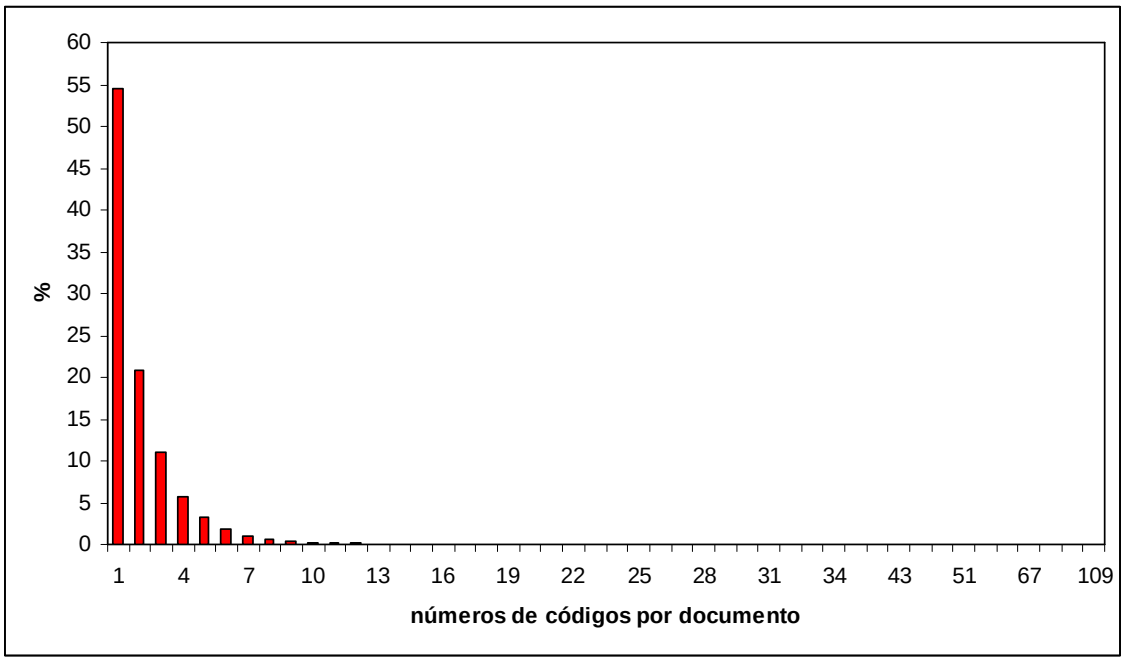

Figura 3-29: Dados de Vitória e BH combinados - Número de códigos por documento

Tabela 3-27: Dados de Vitória e BH combinados - Estatística descritiva para o número de códigos por documento

Mínimo	Máximo	Média	Mediana	Moda	Desvio Padrão
1	109	2,1	1	1	2,0

Verifica-se que, a maior parte, 54,52% num total de 91281 documentos, recebeu apenas um código, o valor máximo de códigos por documento foi de 109 e o número médio de códigos por documento foi aproximadamente igual a 2, neste banco de dados de Vitória e Belo Horizonte.

Tabela 3-28: Dados de Vitória e BH combinados - Distribuição de freqüências por seção

Seção	Freqüência	%
A	490	0,26
B	10	0,01
C	235	0,12
D	11920	6,27
E	178	0,09
F	10133	5,33
G	75443	39,69
H	7045	3,71
I	6888	3,62
J	6323	3,33
K	52323	27,53
L	244	0,13
M	5874	3,09
N	4458	2,35
O	8487	4,47
P	13	0,01
Total	190064	100,00

A Seção G (Comércio; reparação de veículos automotores, objetos pessoais e domésticos, conforme a Tabela CNAE-FISCAL 1.1) foi a mais freqüente, com um percentual de 36,69% de 190064 códigos atribuídos.

Tabela 3-29: Dados de Vitória e BH combinados - Distribuição de freqüências por Divisão

Divisão	Freqüência	%
52	48492	25,51
74	26602	14,00
51	17903	9,42
70	12970	6,82
45	10133	5,33
50	9048	4,76
72	8627	4,54
55	7045	3,71
80	5874	3,09
85	4458	2,35
92	3771	1,98
71	3761	1,98
22	3500	1,84
65	3241	1,71
60	3105	1,63
67	2655	1,40
91	2430	1,28
63	2312	1,22
93	1997	1,05
29	1937	1,02
64	1367	0,72
18	1296	0,68
15	888	0,47
28	866	0,46
36	817	0,43
33	574	0,30
01	437	0,23
66	427	0,22
73	363	0,19
31	294	0,15
90	289	0,15
26	251	0,13
75	244	0,13
17	237	0,12
24	235	0,12
19	189	0,10
40	151	0,08
25	143	0,08
20	140	0,07
34	132	0,07
14	125	0,07
32	118	0,06
13	92	0,05
27	82	0,04
62	81	0,04
21	75	0,04
35	71	0,04
02	53	0,03
30	35	0,02
37	32	0,02
41	27	0,01
61	23	0,01
11	13	0,01
95	13	0,01
05	10	0,01
23	8	0,00
10	5	0,00
Total	190064	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

A maior parte dos códigos, aproximadamente 25%, pertencia à Divisão 52 (Comércio varejista e reparação de objetos pessoais e domésticos) e aproximadamente 14% pertenciam à Divisão 74 (Atividades imobiliárias, aluguéis e serviços prestados às empresas), conforme a Tabela CNAE-FISCAL 1.1, num total de 190064 códigos.

Tabela 3-30: Dados de Vitória e BH combinados - Distribuição de freqüências por Grupo

Grupo	Freqüência	%
524	27985	14,72
523	9625	5,06
741	8634	4,54
704	8612	4,53
749	7250	3,81
511	6923	3,64
552	6491	3,42
742	6019	3,17
522	5756	3,03
452	5059	2,66
514	4671	2,46
851	4133	2,17
809	4122	2,17
502	3451	1,82
503	3136	1,65
602	3091	1,63
722	2745	1,44
527	2535	1,33
521	2428	1,28
222	2244	1,18
744	2211	1,16
713	2047	1,08
454	2039	1,07
930	1997	1,05
919	1982	1,04
672	1968	1,04
659	1924	1,01
516	1922	1,01
703	1711	0,90
515	1704	0,90
701	1649	0,87
501	1635	0,86
299	1634	0,86
725	1613	0,85
455	1565	0,82
513	1524	0,80
729	1500	0,79
926	1468	0,77
801	1357	0,71
923	1316	0,69
181	1234	0,65
745	1211	0,64
723	1206	0,63
221	1191	0,63
721	1148	0,60
702	998	0,53
Total	190064	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Duzentos e doze Grupos foram observados. A tabela acima (Tabela 3-30) apresenta somente os Grupos que apresentam uma porcentagem superior a 0,5%. O Grupo mais freqüente neste banco de dados foi o 524 (Comércio varejista de outros produtos), estando presente em 14,72%, num total de 190064 códigos.

Tabela 3-31: Dados de Vitória e BH combinados - Distribuição de freqüência por Classe

Classe	Freqüência	%
52493	9245	4,86
70408	8612	4,53
74993	6471	3,40
74209	6019	3,17
74160	5670	2,98
52329	4632	2,44
80993	4037	2,12
52442	3796	2,00
50202	3451	1,82
45217	3384	1,78
52418	3262	1,72
51195	3261	1,72
50300	3136	1,65
52450	3122	1,64
55212	3080	1,62
52469	3024	1,59
55220	2879	1,51
52310	2813	1,48
52434	2707	1,42
52426	2503	1,32
74403	2211	1,16
52337	2180	1,15
85138	2120	1,12
51187	2102	1,11
52299	2037	1,07
67202	1968	1,04
60267	1901	1,00
52132	1765	0,93
65994	1747	0,92
22292	1690	0,89
70106	1649	0,87
50105	1635	0,86
72508	1613	0,85
45500	1565	0,82
72907	1500	0,79
91995	1442	0,76
74128	1429	0,75
72290	1389	0,73
72214	1356	0,71
52213	1344	0,71
93025	1287	0,68
52795	1263	0,66
51497	1249	0,66
52710	1233	0,65
74500	1211	0,64
72303	1206	0,63
45411	1205	0,63
52248	1159	0,61
51691	1153	0,61
72109	1148	0,60
70327	1065	0,56
71390	1037	0,55
92312	1025	0,54
85154	1014	0,53
70203	998	0,53
29963	964	0,51
Total	190064	100,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Foram observados 528 diferentes Classes. A tabela acima (Tabela 3-31) apresenta somente as Classes que apresentam uma porcentagem superior a 0,5%. A Classe mais freqüente neste banco de dados foi a 52493 (Comércio varejista de outros produtos não especificados anteriormente), estando presente em 4,86%, num total de 190064 códigos.

Tabela 3-32: Dados de Vitória e BH combinados - Distribuição de freqüências por Subclasse

Subclasse	Freqüência	%
7040800	8612	4,53
7416002	5600	2,95
5232900	4632	2,44
7420902	4171	2,19
5119500	3261	1,72
5249399	2956	1,56
4521701	2954	1,55
5522000	2879	1,51
8099305	2272	1,20
5249303	2126	1,12
5118700	2102	1,11
5020201	2049	1,08
5030003	2005	1,05
5245002	1984	1,04
5521201	1810	0,95
5246902	1785	0,94
5231002	1771	0,93
7010600	1649	0,87
7250800	1613	0,85
5213202	1605	0,84
5242601	1576	0,83
7499399	1534	0,81
7290700	1500	0,79
9199500	1442	0,76
5241804	1407	0,74
5233701	1402	0,74
7229000	1389	0,73
5229999	1384	0,73
7221400	1356	0,71
7440301	1274	0,67
5521202	1270	0,67
8513801	1266	0,67
7230300	1206	0,63
6720201	1197	0,63
5243401	1186	0,62
6599403	1186	0,62
5224800	1159	0,61
7210900	1148	0,60
4541101	1110	0,58
7499307	1069	0,56
7032700	1065	0,56
7499302	1058	0,56
7412801	1057	0,56
7020300	998	0,53
5244201	980	0,52
7499308	951	0,50
Total	190064	100,00

No banco de dados de Vitória-ES foram observadas 764 diferentes Subclasses, no banco de dados de Belo Horizonte-MG foram observadas 1002 diferentes Subclasses, já na combinação dos dois bancos foram observadas 1055 diferentes Subclasses. Cento e duas dessas Subclasses (9,67%) estavam presentes em apenas um documento. De acordo com a tabela acima (Tabela 3-32), pode-se observar que as Subclasses mais freqüentes no banco de dados combinado foram a 7040800 (Condomínios e prédios residenciais ou não) e a 7416002 (Atividades de assessoria em gestão empresarial), com percentual de 4,53% e 2,95% respectivamente, num total de 190064 códigos. A tabela acima (Tabela 3-32) apresenta somente as subclasses que apresentaram freqüência superior a 0,5%, num total de 190064 códigos.

Base 3

Os dados desta base ainda estão sendo disponibilizados.

Base 4

Como mencionado acima, a **BASE 4** será classificada por classificadores manuais, denominados Classificadores Calibrados. Os Classificadores Calibrados são indivíduos treinados em classificação de objetos sociais de acordo com a Tabela CNAE-2.0, com capacidade de classificação e concordância entre eles aferida.

Propomos um **Sistema Online de Avaliação Automática a Distância para Classificadores de Atividades Econômicas (SOAD-CNAE)**, a ser desenvolvido pela aluna de mestrado Márcia Gonçalves de Oliveira (orientada pelo Prof. Elias Silva de Oliveira) em parceria com a equipe estatística do projeto. A seguir apresentamos resumidamente esta proposta.

A proposta de automatizar o processo avaliativo de classificadores de atividades econômicas visa agilizar e tornar o processo de avaliação de classificadores de atividades econômicas independente de fatores como o tempo e o espaço físico que podem dificultar essa avaliação.

O sistema de avaliação online proposto terá como principal função avaliar quão certa uma questão discursiva pode ser considerada. Para essa avaliação, as questões poderão ser sorteadas de uma base de dados e separadas por nível de dificuldade. Depois de selecionar e agrupar as questões de uma avaliação online, programam-se tempos para os classificadores avaliados obterem as soluções de cada grupo (por nível de dificuldade) de questões.

O modelo de avaliação automática dos classificadores de atividades econômicas funcionará da seguinte forma:

Depois de um processo de autenticação, o classificador terá acesso a uma avaliação online.

a) Na página acessada, são informados automaticamente o nome do classificador e o código da prova. As questões são apresentadas ao classificador seguindo uma seqüência por nível de dificuldade. O nível de dificuldade e o número da questão também são mostrados automaticamente.

b) Para resolver as questões de um nível de dificuldade, o classificador tem um tempo estabelecido. É disponibilizado na página um contador automático que possibilita ao classificador verificar o tempo que ainda tem para resolver as questões de determinado nível de dificuldade.

c) O objetivo de cada questão é atribuir códigos de classificação às atividades propostas nas questões. Mas o sistema deve também considerar como o classificador chegou àquele código.

d) Para isso, o classificador pode fazer perguntas. Essas perguntas são avaliadas comparando-as com as perguntas já realizadas pelo IBGE para atribuir um código à atividade econômica descrita como questão da avaliação. Para avaliar as perguntas realizadas pelo classificador, propomos um sistema que detecte automaticamente o índice de similaridade entre as perguntas do classificador e as perguntas já existentes na base fornecida pelo IBGE para classificar esta atividade econômica. Caso essas perguntas sejam similares às perguntas feitas pelo IBGE, o usuário terá essas respostas.

e) O classificador avaliado também poderá fazer consultas por palavras ao sistema de códigos de atividades econômicas do sistema CNAE-2.0.

f) Caso seja necessário inibir o "chute" de códigos por parte do classificador, poderão ser atribuídas penalidades aos códigos errados. Como essa atribuição pode ser difícil, o usuário em vez de "chutar" tem a possibilidade de deixar o código em branco. Caso tenha deixado todos os códigos em branco, o classificador poderá apresentar uma justificativa.

g) As perguntas, consultas por palavras e a justificativa são armazenadas em bancos de dados como informações para avaliação.

3.7.2 Novas Bases Computadas a partir das Bases de Dados de Objetos Sociais de Vitória e Belo Horizonte

Foram observados 1.055 diferentes subclasses nos dados combinados de Vitória/ES e Belo Horizonte/MG. Cento e duas dessas subclasses (9,67%) estavam presentes em apenas um documento. A tabela do Anexo 1 apresenta a distribuição de freqüências para todos os códigos observados. As subclasses mais freqüentes neste banco de dados foram a 7040800 (Condomínios e prédios residenciais ou não) e a 7416002 (Atividades de assessoria em gestão empresarial), com percentual de 4,53% e 2,95% respectivamente. O número total de códigos foi de 190.064, em 91.281 documentos. Assim, observa-se que alguns códigos são muito freqüentes e outros são pouco freqüentes, isto é, existe uma desigualdade na representatividade dos códigos. Visando suavizar essa desigualdade, são propostas novas bases de dados, detalhadas na Tabela 3-33.

Tabela 3-33: Novas Bases de Dados de Vitória e BH

Base	Descrição	Número estimado da amostra
Base VBH1	Nesta base de dados **não** serão incluídos os códigos com freqüência inferior a 100. Para os códigos com freqüência maior do que 100 deverá ser retirada uma amostra aleatória de 100 documentos.	De acordo com a tabela do Apêndice 1, são 312 códigos que apresentam uma freqüência igual ou superior a 100, então o número máximo estimado de documentos incluídos na amostra é igual a 31200 (312x100) (*)
Base VBH2	Serão incluídos todos os documentos relacionados aos códigos com freqüência igual ou inferior a 100. Para os códigos com mais de 100 repetições, deverá ser retirada uma amostra aleatória de 100 documentos.	De acordo com a tabela do Apêndice 1, são 744 códigos que apresentam freqüência igual ou inferior a 100, totalizando 15213 documentos (alguns documentos podem ter recebido mais de um desses códigos). E são 311 códigos com mais de 100 repetições, totalizando uma amostra de 31100 (311*100). Assim o número máximo estimado de documentos incluídos na amostra é igual a 46313 documentos (15213+31100) (**)
Base VBH3	Nesta base **não** serão incluídos os códigos com freqüência inferior a 30. Para os códigos com freqüência maior do que 30 deverá ser retirada uma amostra aleatória de 30 documentos.	De acordo com a tabela do Apêndice 1, são 494 códigos que apresentam uma freqüência igual ou superior a 30, então o número máximo estimado de documentos incluídos na amostra é igual a 14820 (494x30) (***)
Base VBH4	Serão incluídos todos os documentos relacionados aos códigos com freqüência igual ou inferior a 30. Para os códigos com freqüência superior a 30 deverá ser retirada uma amostra aleatória de 30 documentos.	De acordo com a tabela do Apêndice 1, são 569 códigos que apresentam freqüência igual ou inferior a 30, totalizando 5145 documentos (alguns documentos podem ter recebido mais de um desses códigos). E são 486 códigos com mais de 30 repetições, totalizando uma amostra de 14580 (486x30). Assim o número máximo estimado de documentos incluídos na amostra é igual a 19725 documentos (5145+14580)
(*) Um mesmo documento poderá fazer parte da amostra de mais de um código, assim o tamanho da amostra poderá ser inferior ao estimado (31200)		
(**) Um mesmo documento poderá fazer parte da amostra de mais de um código, assim o tamanho da amostra poderá ser inferior ao estimado (46313)		
(***) Um mesmo documento poderá fazer parte da amostra de mais de um código, assim o tamanho da amostra poderá ser inferior ao estimado (14820)		

3.7.3 Métricas de Avaliação para Categorizadores de Texto Multi-Label

Categorização de texto pode ser definida como a tarefa de atribuir categorias (ou rótulos), a partir de um conjunto de categorias predefinidas, a documentos (SEBASTIANI, 2002). Na categorização de texto *multi-label*, uma ou mais categorias podem ser atribuídas a um documento.

Seja **D** um domínio de documentos, $C = \{c_1, \ldots, c_{|C|}\}$ um conjunto de categorias pré-definidas, e $D = \{d_1, \ldots, d_{|D|}\}$ um *corpus* inicial de documentos previamente classificados manualmente por um especialista do domínio em subconjuntos de categorias de *C*. Na aprendizagem *multi-label*, o conjunto de treinamento(-e-validação) $TV = \{d_1, \ldots, d_{|TV|}\}$ é composto por um número de documentos, cada um associado a um subconjunto das categorias de *C*. *TV* é usado para treinar e validar (na verdade, para calibrar parâmetros eventuais de) um sistema de categorização que associa a combinação adequada de categorias às características de cada documento no *TV*. O conjunto de teste $Te = \{d_{|TV+1|}, \ldots, d_{|D|}\}$, por outro lado, é constituído de documentos para os quais as categorias são desconhecidas para o sistema de categorização. Depois de ter sido (calibrado e) treinado com *TV*, o sistema de categorização é utilizado para predizer o conjunto de categorias de cada documento no *Te*.

Um sistema de categorização *multi-label* tipicamente implementa uma função da forma $f : \mathbf{D} \times C \rightarrow R$ que retorna um grau de crença para cada par $\langle d_j, c_i \rangle \in \mathbf{D} \times C$, ou seja, um número entre 0 e 1 que, grosso modo, representa a evidência para o fato de que o documento de teste d_j deve ser categorizado sob a categoria c_i. A função $f(.,.)$ pode ser transformada numa função *ranking* $r(.,.)$, tal que: (i) se $f(d_j, c_i) > f(d_j, c_k)$, então $r(d_j, c_i) < r(d_j, c_k)$; (ii) se $f(d_j, c_i) < f(d_j, c_k)$, então $r(d_j, c_i) > r(d_j, c_k)$; e se $f(d_j, c_i) = f(d_j, c_k)$, então $r(d_j, c_i) = r(d_j, c_k)$.

Seja C_j o conjunto de categorias pertinentes ao documento de teste d_j e C_j^{τ} o conjunto de categorias preditas para d_j. Um sistema de categorização bem sucedido tenderá a posicionar as categorias em C_j em posições mais elevadas no *ranking* do que aquelas não em C_j. As categorias c_i cujo grau de crença é superior a um limiar τ_i são então preditas para o documento de teste d_j, i.e., $C_j^{\tau} = \{c_i \mid f(d_j, c_i) \geq \tau_i\}$. Diferentes limiares τ_i são tipicamente escolhidos para as diferentes categorias c_i.

As métricas utilizadas na literatura para avaliar o desempenho da categorização de texto podem ser classificadas em dois grupos básicos: (i) métricas de avaliação para conjuntos de resposta não-ordenados; e (ii) métricas de avaliação para conjuntos de resposta ordenados. As métricas para conjuntos de resposta não-ordenados avaliam o conjunto exato de categorias, C_j^{τ}, predito para o documento de teste d_j, entre as quais as mais freqüentes são *precision* (MANNING et al., 2008; SEBASTIANI, 2002), *recall* (MANNING et al., 2008; SEBASTIANI, 2002), F_β (MANNING et al, 2008; SEBASTIANI, 2002), e *hamming loss* (SCHAPIRE; SINGER, 1999). As métricas para conjuntos de resposta ordenados avaliam o *ranking* completo derivado da função $f(.,.)$; estas incluem *R-precision* (MANNING et al., 2008), *average precision* (MANNING et al., 2008), *coverage* (SCHAPIRE; SINGER, 2000), *ranking loss* (SCHAPIRE; SINGER, 1999), e *one-error* (SCHAPIRE; SINGER, 1999).

Apresentamos cada uma destas métricas a seguir.

Métricas de Avaliação para Conjuntos de Resposta Não-Ordenados

- Pr$ecision_i^c$, *precision* quanto à categoria c_i, avalia a fração de documentos de teste categorizados sob a categoria c_i que são verdadeiramente associados com c_i:

$$precision_i^c = \frac{\left|D_i^{\tau_i} \cap D_i\right|}{\left|D_i^{\tau_i}\right|}$$

onde D_i é o conjunto de documentos verdadeiramente associados com c_i e $D_i^{\tau_i}$ é o conjunto de documentos categorizados sob c_i.

O valor de $precision_i^c$ pode também ser estimado em termos da tabela de contingência para c_i, mostrada na Tabela 3-34, como:

$$precision_i^c = \frac{TP_i}{TP_i + FP_i}$$

onde FP_i (falsos positivos para c_i) é o número de documentos de teste que foram incorretamente categorizados sob c_i; TN_i (verdadeiros negativos para c_i) é o número de documentos de teste que foram corretamente não categorizados sob c_i; TP_i (verdadeiros positivos para c_i) é o número de documentos de teste que foram corretamente categorizados sob c_i; e FN_i (falsos negativos para c_i)) é o número de documentos de teste que foram incorretamente não categorizados sob c_i.

Tabela 3-34: A tabela de contingência para a categoria c_i

Categoria c_i		Julgamentos do especialista	
		SIM	NÃO
Julgamentos do categorizador	SIM	TP_i	FP_i
	NÃO	FN_i	TN_i

Pode-se calcular a média destes valores relativos à categoria para obter $precision^c$, isto é, um valor global para o conjunto inteiro de categorias do sistema, $|C|$. Para obter uma estimativa para $precision^c$, dois métodos diferentes podem ser adotados:

(i) *macroaveraging*: avalia a média sobre os resultados de diferentes categorias:

$$macro - precision^c = \frac{\sum_{i=1}^{|C|} precision_i^c}{|C|}$$

(ii) *microaveraging*: avalia a soma sobre todas as decisões individuais em termos da tabela de contingência para a categoria c_i:

$$micro - precision^c = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)}$$

Os métodos *macroaveraging* e *microaveraging* podem dar resultados bastante diferentes, especialmente se as diferentes categorias têm generalidades muito diferentes (MANNING et al, 2008; SEBASTIANI, 2002). A habilidade de um categorizador de se comportar bem também mediante categorias com generalidade baixa (i.e., categorias com poucas instâncias de treinamento positivas) será evidenciada por *macroaveraging* e muito menos por *microaveraging*. *Macroaveraging* dá peso igual para cada categoria, enquanto *microaveraging* dá peso igual para cada decisão de categorização. Desta forma, categorias com alta generalidade (i.e., categorias com muitas instâncias de treinamento positivas) dominam aquelas com baixa generalidade em *microaveraging*. Resultados *microaveraged* são, portanto, uma medida de desempenho em categorias com alta generalidade no conjunto de teste. Para avaliar a habilidade de um categorizador se comportar bem também em categorias com baixa generalidade, resultados *microaveraged* devem ser usados.

Quanto maior o valor de *macro-precision*^{c} e *micro-precision*^{c}, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando *macro-precision*^{c} = 1 e *micro-precision*^{c} = 1.

- $\mathrm{Re}call_i^c$, *recall* quanto à categoria c_i, avalia a fração de documentos de teste verdadeiramente associados com a categoria c_i que são categorizados sob c_i:

$$recall_i^c = \frac{\left|D_i^{\tau_i} \cap D_i\right|}{\left|D_i\right|}$$

O valor de $recall_i^c$ pode também ser obtido em termos da tabela de contingência para c_i (veja a Tabela 3-34) como:

$$recall_i^c = \frac{TP_i}{TP_i + FN_i}$$

Estimativas de *macro-recall*^{c} e *micro-recall*^{c} são calculadas como:

$$macro-recall^c = \frac{\sum_{i=1}^{|C|} recall_i^c}{|C|}$$

$$micro-recall^c = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)}$$

Quanto maior o valor de *macro-recall*^{c} e *micro-recall*^{c}, melhor o desempenho do sistema de categorização. O desempenho é perfeito quanto *macro-recall*^{c} = 1 e *micro-recall*^{c} = 1.

- $Macro - F_\beta^c$ avalia a média harmônica ponderada de *macro-precision*^{c} e *macro-recall*^{c}:

$$macro - F_\beta^c = \frac{(\beta^2 + 1) macro - precision^c \times macro - recall^c}{\beta^2 macro - precision^c + macro - recall^c}$$

Nesta fórmula, β pode ser visto como o grau relativo de importância atribuído a *macro-precision*^{c} e *macro-recall*^{c}. Se β = 0 então $macro - F_\beta^c$ coincide com *macro-precision*^{c}, enquanto se β = +∞ então $macro - F_\beta^c$ coincide com *macro-recall*^{c}. Usualmente, o valor β = 1 é usado, que atribui igual importância para *macro-precision*^{c} e *macro-recall*^{c}.

Quanto maior o valor de $macro - F_\beta^c$, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando $macro - F_\beta^c = 1$.

- $Micro - F_\beta^c$ avalia a média harmônica ponderada de *micro-precision*^{c} e *micro-recall*^{c}:

$$micro - F_\beta^c = \frac{(\beta^2 + 1) micro - precision^c \times micro - recall^c}{\beta^2 micro - precision^c + micro - recall^c}$$

Quanto maior o valor de $micro - F_\beta^c$, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando $micro - F_\beta^c = 1$.

- $\Pr ecision_j^d$, *precision* quanto ao documento de teste d_j, avalia a fração de categorias preditas que são pertinentes para o documento de teste d_j:

$$precision_j^d = \frac{\left|C_j^\tau \cap C_j\right|}{\left|C_j^\tau\right|}$$

O valor de $precision_j^d$ pode também ser estimado em termos da tabela de contingência para d_j, como mostrado na Tabela 3-35, como:

$$precision_j^d = \frac{TP_j}{TP_j + FP_j}$$

onde FP_j (falsos positivos para d_j) é o número de categorias que foram incorretamente preditas para d_j; TN_j (verdadeiros negativos para d_j), TP_j (verdadeiros positivos para d_j), e FN_j (falsos negativos para d_j) são definidos de acordo.

Tabela 3-35: A tabela de contingência para o documento de teste d_j

Documento d_j		Julgamentos do especialista	
		SIM	NÃO
Julgamentos do categorizador	SIM	TP_j	FP_j
	NÃO	FN_j	TN_j

Para obter uma estimativa para $precision^d$, isto é, um valor global para o conjunto inteiro de documentos de teste, $|Te|$, os métodos *macroaveraging* e *microaveraging* podem ser adotados:

$$macro - precision^d = \frac{\sum_{j=1}^{|Te|} precision_j^d}{|Te|}$$

$$micro - precision^d = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|} (TP_j + FP_j)}$$

Quanto maior o valor de $macro\text{-}precision^d$ e $micro\text{-}precision^d$, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando $macro\text{-}precision^d = 1$ e $micro\text{-}precision^d = 1$.

Note que *microaveraged precision*, *microaveraged recall* e *microaveraged* F_β dão resultados iguais, independentemente de serem definidas com base na categoria ou com base no documento. Seja $FP_{ij} = 1$ se c_i foi incorretamente predita para d_j, e $FP_{ij} = 0$ caso contrário; e $TP_{ij} = 1$ se c_i foi corretamente predita para d_j, e $TP_{ij} = 0$ caso contrário. Estimativas de *microaveraged precision* com base na categoria, $micro\text{-}precision^c$, e com base no documento, $micro\text{-}precision^d$, podem ser obtidas através da Equação 3-26 e Equação 3-27, respectivamente, como:

$$micro-precision^{c} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)}$$

$$micro-precision^{c} = \frac{\sum_{i=1}^{|C|}\sum_{j=1}^{|Te|} TP_{ij}}{\sum_{i=1}^{|C|}(\sum_{j=1}^{|Te|} TP_{ij} + \sum_{j=1}^{|Te|} FP_{ij})}$$

Equação 3-26

$$micro-precision^{d} = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|}(TP_j + FP_j)}$$

$$micro-precision^{d} = \frac{\sum_{j=1}^{|Te|}\sum_{i=1}^{|C|} TP_{ij}}{\sum_{j=1}^{|Te|}(\sum_{i=1}^{|C|} TP_{ij} + \sum_{i=1}^{|C|} FP_{ij})}$$

Equação 3-27

Como pode ser observado na Equação 3-26 e Equação 3-27, *micro-precision*^{c} é igual a *micro-precision*^{d}. Analogamente, pode-se mostrar que *micro-recall*^{c} é igual a *micro-recall*^{d}, e $micro-F_{\beta}^{c}$ é igual a $micro-F_{\beta}^{d}$.

- $\mathrm{Re}call_j^d$, *recall* quanto ao documento de teste d_j, avalia a fração de categorias pertinentes que são preditas para o documento de teste d_j:

$$recall_j^d = \frac{\left|C_j^{\tau} \cap C_j\right|}{\left|C_j\right|}$$

O valor de $recall_j^d$ pode também ser obtido em termos da tabela de contingência para d_j (veja a Tabela 3-35) como:

$$recall_j^d = \frac{TP_j}{TP_j + FN_j}$$

Estimativas de *macro-recall*^{d} e *micro-recall*^{d} são computadas como:

$$macro-recall^{d} = \frac{\sum_{j=1}^{|Te|} recall_j^d}{|Te|}$$

$$micro - recall^d = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|} (TP_j + FN_j)}$$

Quanto maior o valor de *macro-recall*^{d} e *micro-recall*^{d}, melhor o desempenho do sistema de categorização. O desempenho é perfeito quanto *macro-recall*^{d} = 1 e *micro-recall*^{d} = 1.

- $Macro - F_\beta^d$ avalia a média harmônica ponderada de *macro-precision*^{d} e *macro-recall*^{d}:

$$macro - F_\beta^d = \frac{(\beta^2 + 1) macro - precision^d \times macro - recall^d}{\beta^2 macro - precision^d + macro - recall^d}$$

Quanto maior o valor de $macro - F_\beta^d$, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando $macro - F_\beta^d = 1$.

- $Micro - F_\beta^d$ avalia a média harmônica ponderada de *micro-precision*^{d} e *micro-recall*^{d}:

$$micro - F_\beta^d = \frac{(\beta^2 + 1) micro - precision^d \times micro - recall^d}{\beta^2 micro - precision^d + micro - recall^d}$$

Quanto maior o valor de $micro - F_\beta^d$, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando $micro - F_\beta^d = 1$.

- *Hamming-loss*$_j$ avalia quantas vezes o documento de teste d_j é classificado erroneamente (i.e., uma categoria não pertencente ao documento é predita ou uma categoria pertencente ao documento não é predita), normalizada pelo número total de categorias no sistema:

$$ham\min g - loss_j = \frac{|C_j^\tau \Theta C_j|}{|C|}$$

onde Θ é a diferença simétrica entre o conjunto de categorias preditas, C_j^τ, e o conjunto de categorias pertinentes C_j de d_j.

Para os |*Te*| documentos de teste, o desempenho global é obtido ao calcular a média dos resultados para todos os |*Te*| documentos, isto é,

$ham\min g-loss=\frac{1}{|Te|}\sum_{j=1}^{|Te|}ham\min g-loss_j$. Quanto menor o valor de *hamming-loss*, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando *hamming-loss* = 0.

- *R-hamming-loss*$_j$ avalia quantas vezes o documento de teste d_j é classificado erroneamente, normalizada pelo número total de categorias pertinentes:

$$R-ham\min g-loss_j=\frac{\left|C_j^{\tau}\Theta C_j\right|}{\left|C_j\right|}$$

O desempenho global é calculado como $R-ham\min g-loss=\frac{1}{|Te|}\sum_{j=1}^{|Te|}R-ham\min g-loss_j$. Quanto menor o valor de *R-hamming-loss*, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando *R-hamming-loss* = 0.

Métricas de Avaliação para Conjuntos de Resposta Ordenados

- $R-\Pr ecision_j^d$ avalia o valor de *precision* quanto ao documento de teste d_j, $precision_j^d$, computado depois de truncar o *ranking* de categorias para o documento de teste d_j na posição $k=\left|C_j\right|$:

$$R-precision_j^d=\frac{\left|C_j^{|C_j|}\cap C_j\right|}{\left|C_j^{|C_j|}\right|}$$

onde $C_j^{|C_j|}$ é o conjunto de categorias que vão do topo do *ranking* até a posição $k=\left|C_j\right|$ do *ranking*.

O desempenho global é dado por $R-precision^d=\frac{1}{|Te|}\sum_{j=1}^{|Te|}R-precision_j^d$. Quanto maior o valor de *R-precision*^{d}, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando *R-precision*^{d} = 1.

- $Avg-precision_j^d$ avalia a média dos valores de *precision* quanto ao documento de teste d_j, $precision_j^d$, computados depois de truncar o *ranking* de categorias para o documento de teste d_j depois de cada categoria $c_i \in C_j$:

$$avg-precision_j^d = \frac{1}{|C_j|}\sum_{k=1}^{m}\frac{|C_j^k \cap C_j|}{|C_j^k|}$$

onde *m* é o número de posições no *ranking* que têm pelo menos uma categoria $c_i \in C_j$ para d_j, e C_j^k é o conjunto de categorias que vão do topo do *ranking* até a posição *k* do *ranking*. Se existe uma categoria $c_i \in C_j$ na posição *k* e $f(d_j, c_i) = 0$ então $\frac{|C_j^k \cap C_j|}{|C_j^k|} = 0$.

O desempenho global é computado como $avg-precision^d = \frac{1}{|Te|}\sum_{j=1}^{|Te|} avg-precision_j^d$. Quanto maior o valor de *avg-precision*^{d}, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando *avg-precision*^{d} = 1.

- *Coverage*$_j$ mede até onde precisamos descer no *ranking* de categorias para o documento de teste d_j a fim de cobrir todas as categorias pertinentes do documento:

$$coverage_j = \max_{c_i \in C_j} r(d_j, c_i) - 1$$

onde $\max_{c_i \in C_j} r(d_j, c_i)$ retorna a posição máxima do conjunto de categorias pertinentes de d_j.

O desempenho global é obtido por $coverage = \frac{1}{|Te|}\sum_{j=1}^{|Te|} coverage_j$. Quanto menor o valor de *coverage*, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando $coverage = \frac{1}{|Te|}\sum_{j=1}^{|Te|}\left(|C_j| - 1\right)$.

- *Ranking-loss*$_j$ avalia a fração de pares de categorias (<c_i, c_k>, para os quais $c_i \in C_j$ e e $c_k \in \overline{C}_j$) que estão reversamente ordenados ($f(d_j, c_i) \leq f(d_j, c_k)$) no *ranking* de categorias para o documento de teste d_j:

$$ranking-loss_j = \frac{\left|\left\{(c_i, c_k) \in C_j \times \overline{C}_j \,|\, f(d_j, c_i) \leq f(d_j, c_k)\right\}\right|}{|C_j||\overline{C}_j|}$$

onde $\overline{C}_j$ é o conjunto complementar de C_j em *C*.

O desempenho global é calculado como $ranking-loss = \frac{1}{|Te|}\sum_{j=1}^{|Te|} ranking-loss_j$.

Quanto menor o valor de *ranking-loss*, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando *ranking-loss* = 0.

- *One-error*$_j$ avalia se as categorias no topo do *ranking* estão presentes no conjunto de categorias pertinentes do documento de teste d_j:

  one-error$_j$ = 0, if $c_i \in C_j$, $f(d_j, c_i) > 0$, $\forall c_i \in \arg\max_{c_i \in C} f(d_j, c_i)$
  one-error$_j$ = 1, otherwise

  onde $\arg\max_{c_i \in C} f(d_j, c_i)$ retorna as categorias no topo do *ranking* para d_j.

  O desempenho global é dado por $one-error = \frac{1}{|Te|}\sum_{j=1}^{|Te|} one-error_j$. Quanto menor o valor de *one-error*, melhor o desempenho do sistema de categorização. O desempenho é perfeito quando *one-error* = 0.

3.8 Meta Física 3.2/2007 – Avaliação Estatística dos Mecanismos de Codificação Desenvolvidos

3.8.1 Conceitos em Planejamento de Experimento Estatístico

Experimentos aleatórios são procedimentos estímulo-resposta a que submete-se um objeto em estudo que, ao serem repetidos sob as mesmas condições, não produzem as mesmas respostas. Por essa razão, o planejamento de experimentos aleatórios é uma atividade necessária para se descobrir informações fidedignas, apesar das respostas diferentes a cada repetição sobre um processo ou sistema em particular. A principal vantagem do planejamento é a economia de tempo, custos e a redução da variabilidade nos resultados, o que permite conhecer melhor o objeto estudado. Este conhecimento é tipicamente consolidado na forma de um modelo do objeto em estudo (MONTGOMERY, 2001).

Todo planejamento deve respeitar três princípios básicos, que são:

a) **Repetição**: ao se repetir o experimento acrescenta-se aos modelos informações sobre a variabilidade intrínseca do objeto de estudo;

b) **Aleatorização**: quando na presença de um grupo de estímulos com as mesmas características, a escolha ao acaso elimina possíveis viéses de escolha.

c) **Formação de estratos**: a presença de subgrupos que possam trazer variabilidade à resposta de interesse implica na formação de estratos *a priori.*

Para a perfeita execução de um planejamento de experimentos é essencial definir a unidade experimental (ente do qual serão extraídas as informações) e o que será observado (variáveis). Além disso, se forem realizadas comparações entre grupos ou métodos, defini-los claramente.

3.8.2 Índice Para Medir Concordância

3.8.2.1 Concordância entre Métodos para Dados Qualitativos: KAPPA

Para descrever a intensidade da concordância entre dois ou mais métodos de classificação, pode ser utilizado o índice Kappa, proposto por Cohen (COHEN,1960), que é um teste não-paramétrico baseado no número de respostas concordantes, ou seja, no número de casos cujo resultado é o mesmo entre os métodos. O Kappa mede o grau de concordância além do que seria esperado tão somente pelo acaso. Esta medida de concordância tem como valor máximo o 1 (um), o qual representa a total concordância e os valores próximos ou abaixo de zero, indicam nenhuma concordância. Um eventual valor de Kappa menor que zero, negativo, indica que a concordância encontrada foi menor do que aquela esperada por acaso, sugere, portanto, discordância, mas seu valor não tem interpretação como intensidade de discordância.

Para avaliar se a concordância é razoável, pode-se realizar um teste estatístico para analisar a significância do Kappa. Neste caso, a hipótese testada é se o Kappa é igual a zero, o que indicaria concordância nula, ou se ele é maior do que zero, concordância maior do que o

acaso (teste unicaudal: H_0: Kappa = 0; H_1: Kappa > 0). No caso de rejeição da hipótese (Kappa = 0) temos a indicação de que a medida de concordância é significantemente maior do que zero, ou seja, existe alguma concordância. Isto não significa necessariamente que a concordância seja alta, cabe ao pesquisador avaliar se a medida obtida é satisfatória ou não. Landis e Koch (LANDIS; KOCH, 1977) sugerem a seguinte interpretação:

Tabela 3-36: Interpretação do índice de concordância Kappa (LANDIS; KOCH, 1977)

Valores de Kappa	Interpretação
< 0,00	Não concordam
0,00 a 0,19	Concordância baixa
0,20 a 0,39	Concordância justa (balanceada)
0,40 a 0,59	Concordância moderada
0,60 a 0,79	Concordância boa
0,80 a 1,00	Concordância quase perfeita

Essa avaliação de concordância através do Kappa é utilizada quando as escalas são categóricas e sempre quando estamos comparando dois ou mais métodos.

Cálculos de concordância entre modelos são importantes na medida em que as eventuais discordâncias possam ser identificadas, observando-se em qual momento específico se deu à discordância, e caracterizando sua natureza.

O cálculo da medida Kappa deve ser realizado através de uma tabela comparativa, como, por exemplo, a descrita a seguir:

Tabela 3-37: Tabela comparativa para o cálculo de Kappa

Avaliador 1 (resposta Positiva)	**Avaliador 2 (resposta positiva)**		**Total**
	sim	não	
Sim	a	b	a + b
Não	c	d	c + d
Total	a + c	b + d	N

a e d = número de documentos com concordância das categorias sim e não, respectivamente, para os dois avaliadores;

b e c = número de documentos com discordância entre os dois avaliadores;

De acordo com Cohen (COHEN, 1960), o coeficiente Kappa pode ser calculado da seguinte forma:

$$Coeficiente\ Kappa = \frac{Concordância\ observada - Concordância\ Esperada}{N - Concordância\ Esperada}$$

onde, *N* é o total de diagnósticos realizados.

Concordância Observada = Total de diagnósticos coincidentes. É a soma de todos os valores da diagonal da tabela de concordância observada.

Considerando uma tabela com *i* linhas e *j* colunas, onde para cada célula da tabela com uma freqüência observada, deve ser calculada uma freqüência esperada, a freqüência esperada de uma célula *Cij* é dada por:

$$Freqüência\ Esperada\ de\ Cij = \frac{Total\ da\ linha\ i \times Total\ da\ coluna\ j}{N}$$

Concordância Esperada = Soma de todas as freqüências esperadas da diagonal da tabela de concordância esperada.

Para comparar os acertos e os erros de cada método em relação ao codificador manual pode-se calcular a concordância percentual, que é a relação entre os diagnósticos concordantes e todos os diagnósticos dados.

$$Concordância\ percentual = \frac{Concordância\ Observada}{N}$$

3.8.2.2 Concordância entre Métodos para Dados Quantitativos: Coeficiente de Correlação

O Coeficiente de correlação linear (*r*) é uma medida do grau de associação entre duas características *x* e *y* a partir de uma série de observações. O coeficiente de correlação de Pearson (TRIOLA, 1998) é dado por:

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}$$

Onde *x* e *y* denotam o resultado para cada avaliador, subtraídos das respectivas médias.

Interpretação do coeficiente de correlação

O coeficiente de correlação é um valor entre –1 e +1, sendo que *r* = 0 corresponde à não associação entre as variáveis, *r* = 1, ou r = -1 corresponde a relação linear perfeita, assim,

$$\text{Valores de r} \begin{Bmatrix} \text{negativos} \\ \text{positivos} \end{Bmatrix} \text{ indicam uma associação} \begin{Bmatrix} \text{negativa} \\ \text{positiva} \end{Bmatrix}$$

Dizemos que existe correlação linear positiva quando *r* > 0, ou seja, à medida que *x* cresce *y* também cresce. A correlação linear negativa ocorre quando *r* < 0, e nesse caso à medida que *x*

cresce y decresce (em média). Quanto maior o valor de *r* (positivo ou negativo), mais forte é a associação.

A tabela a seguir fornece um guia de como podemos descrever a correlação em categorias. É claro que as interpretações dependem de cada contexto particular.

Tabela 3-38: Interpretação do índice *r* de correlação (SHIMAKURA, 2006)

Valor de *r* (+ ou -)	Interpretação
0,00 a 0,19	Correlação bem fraca
0,20 a 0,39	Correlação fraca
0,40 a 0,69	Correlação moderada
0,70 a 0,89	Correlação forte
0,90 a 1,00	Correlação muito forte

3.8.3 Teste para Comparação de Dois Modelos

Empregam-se testes estatísticos de duas amostras quando se deseja determinar se dois tratamentos são diferentes, ou se um tratamento é "melhor" do que o outro. Na comparação de dois grupos podem-se observar diferenças significativas que não são resultados do tratamento aplicado. Por exemplo, pode-se querer comparar dois métodos de ensino, aplicando um método a um grupo de alunos, e outro método a outro grupo diferente de alunos. Se um dos grupos conta com estudantes mais inteligentes ou mais bem motivados, os resultados obtidos pelos dois grupos, após a aplicação dos diferentes métodos de ensino, não refletirão fielmente a eficiência relativa dos dois métodos, porque outras variáveis estão contribuindo para as diferenças observadas nos resultados.

Uma das maneiras de superar a dificuldade proveniente de diferenças intrínsecas entre dois grupos consiste em utilizar na pesquisa duas amostras relacionadas. Esta relação pode ser obtida utilizando cada elemento como seu próprio controle, ou seja, o elemento é submetido a ambos os tratamentos. No exemplo citado acima, o emparelhamento exigiria que se formassem pares de estudantes de maneira que cada par fosse composto de dois estudantes de inteligência e motivação iguais (SIEGEL, 1988).

3.8.3.1 Testes de Comparação de Duas Médias

O teste de duas médias é realizado para se comparar as médias de duas populações a partir da análise das médias de suas amostras. Em geral fazem-se os testes sobre a diferença entre duas médias populacionais:

$$H_0 : \mu_1 - \mu_2 = \mu_D$$

sendo na maioria dos casos $\mu_D = 0$, o que significa que testa-se a igualdade entre as médias

$$H_0 : \mu_1 = \mu_2$$

As Hipóteses Nula e Alternativa do teste são as seguintes:

Teste bilateral:

$$H_0 : \mu_D = 0$$

$$H_1 : \mu_D \neq 0$$

Teste unilateral à esquerda:

$$H_0 : \mu_D \geq 0$$

$$H_1 : \mu_D < 0$$

Teste unilateral à direita:

$$H_0 : \mu_D \leq 0$$

$$H_1 : \mu_D > 0$$

OBS.: No presente estudo, na comparação dos modelos será considerado o teste bilateral.

Na comparação das médias consideram-se dois casos: dados emparelhados (populações correlacionadas) e dados não emparelhados (populações não correlacionadas). No trabalho, será considerada a comparação de dados emparelhados, uma vez que, os modelos propostos serão testados com o mesmo conjunto de documentos.

3.8.3.2 Duas Médias Pareadas: Teste t - Student

Faz-se testes de comparações de médias para dados emparelhados quando as amostras são relacionadas duas a duas de acordo com algum critério que fornece uma influência entre os vários pares e sobre os valores de cada par. Para cada par definido, o valor da primeira amostra está associado ao valor da segunda amostra (MORETTIN, 1999).

Sejam duas amostras $X_1, X_2, ..., X_n$ e $Y_1, Y_2, ..., Y_n$, sendo as observações pareadas, ou seja, as duas possam ser consideradas uma amostra de pares, $(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$. Definindo-se a v.a. $D = X - Y$, tem-se a amostra $D_1, D_2, ..., D_n$, resultante das diferenças entre os valores de cada par.

Para a aplicação do teste deve-se supor que ambas as amostras são provenientes de populações com distribuição normal, assim, a variável D, supostamente, também tem distribuição normal $N\left(\mu_D, \sigma_D^2\right)$. Daí pode-se deduzir que:

$$\overline{D} = \frac{\sum_{i=1}^{n} D_i}{n} = \frac{\sum_{i=1}^{n} \left(X_i - Y_i\right)}{n} = \overline{X} - \overline{Y}$$

Terá distribuição $N\left(\mu_D, \frac{\sigma_D^2}{n}\right)$

Considere $s_D^2 = \frac{\sum_{i=1}^{n}\left(D_i - \overline{D}\right)^2}{n-1}$

Então a estatística de teste será dada por: $t = \frac{\overline{D} - \mu_D}{s_{\overline{D}}}$

sendo, $s_{\overline{D}} = \frac{s_D}{\sqrt{n}}$

onde *t* tem distribuição *t*-Student com n-1 graus de liberdade.

A regra do teste (teste bilateral) é então rejeitar H_0 se,

$$|t| \geq t_{n-1,\,\frac{\alpha}{2}}$$

onde $t_{n-1,\,\frac{\alpha}{2}}$ é obtido da tabela da distribuição *t*-student, considerando (n-1) graus de liberdade e tomando α como o nível de significância do teste.

Outra maneira de tomar a decisão final sobre a hipótese nula é comparando o valor-p com um valor pré-fixado (nível de significância), usualmente 0,05. Quando o valor-p é menor que este ponto de corte, o resultado é chamado estatisticamente significante e caso contrário é dito não significante.

O valor-p (ou valor de probabilidade) é a probabilidade de obter um valor da estatística amostral de teste no mínimo tão extremo como o que resulta dos dados amostrais, na suposição da hipótese nula ser verdadeira, ou seja, é a probabilidade de ter observado os dados quando a hipótese nula é verdadeira.

No caso do teste bilateral, o valo-p é a duas vezes a área da estatística de teste.

Exemplo: Suponha que se deseja comparar os modelos 1 e 2 e que a métrica utilizada tenha sido *Hamming Loss* (*hloss*$_j$). A tabela a seguir ilustra algumas situações de aplicação de tais modelos.

Tabela 3-39: Comparação de modelos – teste t

d_j	C_j	P_j (Modelo 1)	P_j (Modelo 2)	X_j	Y_j	D_j
d_1	A B C	A	A B C	0,02	0,00	0,02
d_2	A B C D	A B C D H I	A B D	0,04	0,02	0,02
d_3	D E	D F I	D E	0,06	0,00	0,06
…	…	…	…	…	…	…
d_{50}	F	F G	F	0,02	0,00	0,02

Sendo

- d_j: j-ésimo documento;
- C_j: conjunto de códigos apropriados para o j-ésimo documento;
- P_j (Modelo 1): conjunto de códigos preditos pelo Modelo 1 para o j-ésimo documento;
- P_j (Modelo 2): conjunto de códigos preditos pelo Modelo 2 para o j-ésimo documento;
- X_j: resultado da métrica obtida para Modelo 1;
- Y_j: resultado da métrica obtida pelo Modelo 2;
- D_j: diferença entre X_j e Y_j.

Para o exemplo as hipóteses a serem testadas são:

$$H_0 : \mu_{modelo1} - \mu_{modelo2} = \mu_D = 0$$
$$H_1 : \mu_{modelo1} - \mu_{modelo2} = \mu_D \neq 0$$

Suponha que:

$$\alpha = 0{,}05$$
$$\overline{D} = 0{,}03$$
$$s_D = 0{,}02$$
$$s_{\overline{D}} = \frac{s_D}{\sqrt{n}} = 0{,}003$$

Assim,

$$t = \frac{\overline{D} - \mu_D}{s_{\overline{D}}} = \frac{0{,}03 - 0}{0{,}003} = 10$$

$$t_{n-1;\,\frac{\alpha}{2}} = t_{49;\,0{,}025} = 2{,}010$$

Como $t > t_{49;\,0{,}025}$ rejeita-se H_0, isto é, rejeita-se a hipótese que os modelos são iguais.

Considerando a decisão através do valor-p tem-se:

$$valor - p = P(|t| \geq 10) \cong 0{,}000$$

Como o valor-p < 0,05 então rejeita-se H_0.

3.8.3.3 Teste de Duas Proporções: Teste McNemar

O teste McNemar deve ser utilizado quando a variável sob estudo é dicotômica (do tipo zero e um) e se há dependência entre as amostras (amostras pareadas), A Filosofia do teste de

McNemar é a seguinte: Uma amostra A1, submetida a um tratamento T1, e o seu resultado medido. Posteriormente, essa mesma amostra, chamada agora de A2, é submetida a um segundo tratamento T2, medindo-se o seu resultado pela mesma variável usada no primeiro tratamento.

No presente estudo será utilizado o teste McNamer para verificar se existe diferença significativa entre dois métodos (a comparação será realizada dois a dois) utilizando-se a métrica *One-error*.

Comparando-se o resultado dos dois métodos em cada elemento da amostra, podem ocorrer 4 alternativas, conforme tabela abaixo:

Tabela 3-40: Exemplificação da comparação de dois métodos

Método 1	Método 2		Total
	Presente	Ausente	
Presente	*k*	*r*	*n1*
Ausente	*s*	*l*	*n2*
Total	m_1	m_2	N

Se p_1 e p_2 são probabilidades de sucesso nos grupos método 1 e método 2, respectivamente, a hipótese de interesse é:

$$H_0 : p_1 = p_2 \quad \text{versus} \quad H_1 : p_1 \neq p_2$$

Os pares que produziram presente ou ausente nos dois métodos não contêm informação para discriminar p_1 e p_2. Se H_0 é verdadeira, ou seja, os dois métodos são equivalentes, as discordâncias observadas são frutos do acaso. Olhando a tabela acima, se *r* e *s* têm valores semelhantes, sob H_0 espera-se a metade do número de discordâncias $\left(\frac{r+s}{2}\right)$. A hipótese H_0 deve, portanto, ser rejeitada se a distância entre os valores observados e esperados for grande (SOARES; SIQUEIRA, 1999).

Usando a correção de continuidade (esta é necessária porque se utiliza uma distribuição contínua, qui-quadrado, para aproximar uma distribuição discreta), a estatística de teste é dada por:

$$\chi^2_{McN} = \frac{\left(\left|r - \frac{r+s}{2}\right| - \frac{1}{2}\right)^2}{\frac{r+s}{2}} + \frac{\left(\left|s - \frac{r+s}{2}\right| - \frac{1}{2}\right)^2}{\frac{r+s}{2}} = \frac{\left(|r-s|-1\right)^2}{r+s}$$

O teste consiste em rejeitar H_0 quando

$$\chi^2_{McN} = \frac{\left(|r-s|-1\right)^2}{r+s} > \chi^2_{1,1-\alpha}$$

Onde $\chi^2_{1,1-\alpha}$ é o percentil de ordem 1-α da distribuição qui-quadrado com 1 grau de liberdade.

3.9 Meta Física 4.1/2007 – Realização de Seminários de Acompanhamento e Avaliação

Conforme previsto no projeto, foram realizados dois seminários de acompanhamento e avaliação. A agenda destes seminários é como apresentado abaixo. Os slides das apresentações listadas nas duas agendas acompanham este relatório, em formato eletrônico, no DVD anexo.

1º. Seminário de Acompanhamento e Avaliação do Projeto Classificação Automática em CNAE-Fiscal

Dias 14 e 15 de Dezembro de 2007, Novotel, Vitória-ES

Agenda:

Dia 14 de Dezembro - Sexta

14:00 – Alberto Ferreira De Souza

– *Abertura do encontro*

Apresentações Técnicas

14:20 – Fernando Líbio Leite Almeida
– *Arquitetura do SCAE*

15:00 – Alberto Ferreira De Souza, Claudine Santos Badue Gonçalves
– *Categorização de Texto com Redes Neurais Sem Peso VG-RAM*

15:40 – Elias de Oliveira
– *Classificação de Texto com Modelos Vetoriais*

16:20 – Coffee Break

16:40 – Felipe França
– *Classificação de Texto com Redes Neurais Sem Peso*

17:20 – Priscila Machado Vieira Lima
– *Classificação de Texto com Redes Neurais Sem Peso*

18:00 – Hannu Tapio Ahonen
– *Classificação de Texto com Redes Bayesianas*

19:00 – Encerramento 1º dia

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Dia 15 de Dezembro - Sábado

Apresentações Técnicas

08:00 – Eliana Zandonade
– *Criação de uma base de dados representativa de codificações CNAE-Fiscal corretas*

Discussões Técnicas

09:00 – Plenária
– *Novas alternativas para a categorização de texto*

10:00 – Coffee Break

10:20 – Plenária
– *Composição de classifcadores já implementados*

Detalhamento do Cronograma do Segundo Ano

11:20 – Plenária

13:00 – Encerramento

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

2º. Seminário de Acompanhamento e Avaliação do Projeto Classificação Automática em CNAE-Fiscal

Dias 11 e 12 de Abril de 2008, Bristol La Residence, Vitória-ES

Agenda:

Dia 11 de Abril - Sexta

14:00 – Alberto Ferreira De Souza
– Abertura do encontro

Apresentações Técnicas

14:20 – Bruno Zanetti Melotti
– Implementação TFIDF e seus benefícios na categorização de texto com Vector Space e Redes Neurais Sem Peso

15:00 – Felipe Thomaz Pedroni
– Novas funcionalidades adicionadas ao DB_CORE: facilidade de filtro (stemming, retiradas de acentos, retirada de plurais) e correção ortográfica

15:40 – Hannu Tapio Ahonen
– Mecanismo de codificação baseado em redes Bayesianas –BN_CORE

16:20 – Coffee Break

16:40 – Claudine Santos Badue Gonçalves
– Mecanismo de codificação baseado em redes neurais artificiais sem peso com correlação de dados de saída – WNN_COR_CORE

17:20 – Charles Bezerra do Prado
– Mecanismo de composição dos resultados da codificação através de neurais artificiais, redes Bayesianas e Modelo Vetorial em uma única codificação, mais robusta

18:00 – Eliana Zandonade
– Criação de benchmarking para realização de comparações entre os métodos

18:40 – Valmir Carneiro Barbosa
– Agenda de pesquisa: Propriedades de Knowledge Correlated VG-RAM WNNs, Heurísticas de Busca na Memória de Neurônios VG-RAM, Conhecimento Acerca da Categorização Multi-Label de um Conjunto de Documentos em Função de uma Base de Dados de Categorizações Existentes

19:20 – Encerramento 1º dia

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Dia 12 de Abril - Sábado

Apresentações Técnicas

08:00 – Fernando Líbio Leite Almeida
– *O protótipo do SCAE Fiscal*

08:40 – Eliana Zandonade
– *Avaliação estatística dos mecanismos de codificação desenvolvidos*

09:20 – Wagner Meira Jr.
– *Agenda de pesquisa: Categorizador SVM Multi-label, e Técnicas de Datamining na Categorização de Texto, Alternativas para Novos Categorizadores e Composição de Categorizadores*

Discussões Técnicas

10:00 – Coffee Break

10:20 – Plenária
– *Metodologia de avaliação da corretude da codificação humana*

Revisão do Detalhamento do Cronograma do Segundo Ano

11:20 – Plenária

13:00 – Encerramento

4 Outras Realizações Técnico-Científicas

O envolvimento dos pesquisadores principais em outras atividades técnicas e científicas de interesse do Projeto SCAE é aqui relatado.

4.1 Organização e Participação em Eventos Científicos

Algumas oportunidades de participação em eventos científicos, nacionais e internacionais, foram aproveitadas no decorrer do Projeto SCAE. É aqui relatada a proposta de organização da segunda edição do workshop em categorização inteligente de textos (WITCC 2008), aceito como parte integrante de evento internacional na área (ISDA 2008, Taiwan). As participações dos pesquisadores em Comitês de Programa, neste e em outros eventos, também são relacionadas.

4.1.1 Proposta Aceita de Workshop junto ao ISDA 2008

Estabelecido o interesse na organização de um evento científico cujo tema específico fosse compatível com o Projeto SCAE, foi organizado o *1st Workshop on Intelligent Text Categorization and Clustering — WITCC 2007*, associado ao *7th International Conference on Intelligent Systems Design and Applications — ISDA 2007*, Rio de Janeiro. Em 2008 renovamos nossa proposta de organização e tivemos aprovada a realização do *2nd Workshop on Intelligent Text Categorization and Clustering — WITCC 2008* (http://www.cos.ufrj.br/~felipe/ISDA2008_WITCC.html), associado ao *8th International Conference on Intelligent Systems Design and Applications — ISDA 2008.* O referido evento mantém a visibilidade internacional dada (i) a qualidade histórica de suas publicações, (ii) pela qualidade do conjunto de promotores desta edição do ISDA (http://bit.kuas.edu.tw/~isda08/sponsors.html): (iii) pela editoração dos anais com visibilidade internacional (http://ieeexplore.ieee.org/). Coordenado pelos professores Felipe M. G. França e Alberto F. De Souza, todos os pesquisadores do Projeto SCAE, além de outros pesquisadores internacionais e nacionais, entre eles os Prof. Jun Okamoto, USP, participaram dos Comitês de Programa do *WITCC 2008* e *ISDA 2008.*

4.1.2 Participações em Comitês de Programa

Será relacionada aqui a participação dos pesquisadores principais do Projeto SCAE em Comitês de Programa de eventos científicos em áreas afins.

4.1.2.1 SBIA 2008

O professor Felipe M. G. França é membro do Comitê de Programa do *19th Brazilian Symposium on Artificial Intelligence — SBIA 2008* (http://www.sbia2008.ufba.br/sbia.php). A professora e Priscila M. V. Lima também atuou como revisora dos trabalhos submetidos.

4.1.2.2 SBRN 2008

O professor Felipe M. G. França é membro do Comitê de Programa do *10th Brazilian Symposium on Artificial Neural Networks — SBRN 2008* (http://www.sbia2008.ufba.br/sbrn.php). Os professores Alberto F. Souza, Elias Oliveira e Priscila M. V. Lima, além do pesquisador Charles B. Prado, também atuaram como revisores dos trabalhos submetidos.

4.1.2.3 SBPO 2008

Os professores Felipe M. G. França e Priscila M. V. Lima são membros do Comitê de Programa do *XL Simpósio Brasileiro de Pesquisa Operacional — SBPO 2008* (http://www.ufpb.br/sbpo2008/).

4.1.2.4 ISDA 2008

Além da organização do *WITCC 2008*, os pesquisadores Charles B. Prado e Claudine Badue, além de todos os pesquisadores professores do Projeto SCAE, participaram como revisores de trabalhos submetidos não somente ao *WITCC 2008*, mas também de outros trabalhos na área de Inteligência Computacional submetidos ao *ISDA 2008*. Os professores Felipe M. G. França, Alberto F. De Souza e Priscila M. V. Lima são membros do Comitê de Programa do *ISDA 2008* (http://bit.kuas.edu.tw/~isda08/).

4.2 Publicações

1. Nedjah, N. (Org.) ; Mourelle, L. M. (Org.) ; Kacprzyk, J. (Org.) ; Souza, A. F. (Org.) ; França, F. M. G. (Org.) . *Intelligent Text Categorization and Clustering*. 1. ed. Berlin: Springer, March 2009. v. 1. 133 p. (to be Published: http://www.springer.com/March new+%26+forthcoming+titles+%28default%29/book/978-3-540-85643-6).
2. Alberto F. De Souza, Claudine Badue, Bruno Zanetti Melotti, Felipe T. Pedroni, Fernando Líbio L. Almeida. *Improving VG-RAM WNN Multi-label Text Categorization via Label Correlation*. In: 8th International Conference on Intelligent System Design and Applications (ISDA'08), 2nd Workshop on Intelligent Text Categorization and Clustering (WITCC 2008). Kaohsiung City, Taiwan, 2008 (Accepted).
3. Charles B. Prado, Felipe M. G. França, Ramon Diacovo, Priscila M. V. Lima. *The Influence of Order on a Large Bag of Words*. In: 8th International Conference on Intelligent System Design and Applications (ISDA'08), 2nd Workshop on Intelligent Text Categorization and Clustering (WITCC 2008). Kaohsiung City, Taiwan, 2008 (Accepted).
4. Elias Oliveira, Patrick M. Ciarelli, Claudine Badue. *A Comparison Between a kNN based Approach and a PNN Algorithm for a Multi-Label Classification Problem*. In: 8th International Conference on Intelligent System Design and Applications (ISDA'08), 2nd Workshop on Intelligent Text Categorization and Clustering (WITCC 2008). Kaohsiung City, Taiwan, 2008 (Accepted).
5. Claudine Badue, Felipe Pedroni, Alberto F. De Souza. *Multi-Label Text Categorization using VG-RAM Weightless Neural Networks*. In: Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN'08). Salvador, Bahia, Brazil, 2008(Accepted).
6. Elias Oliveira, Patrick M. Ciarelli, Alberto F. De Souza, Claudine Badue. *Using a Probabilistic Neural Network for a Large Multi-Label Problem*. In: Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN'08). Salvador, Bahia, Brazil, 2008

(Accepted).

7. Alberto F. De Souza, Claudine Badue, Felipe Pedroni, Elias Oliveira, Stiven S. Dias, Hallysson Oliveira, Sotério F. de Souza. *Face Recognition with VG-RAM Weightless Neural Networks*. In: Proceedings of the 18th International Conference on Artificial Neural Networks (ICANN'08). Prague, Czech Republic, 2008 (Accepted).
8. Alberto F. De Souza, Felipe Pedroni, Elias Oliveira, Patrick M. Ciarelli, Wallace F. Henrique, Lucas Veronese, Claudine Badue. *Automated Multi-label Text Categorization with VG-RAM Weightless Neural Networks*. *Neurocomputing*, Elsevier Science. 2008 (Accepted).
9. Priscila M. V. Lima, Miriam Mariela M. Morveli-Espinoza, Glaucia C. Pereira, Talita O. Ferreira, Felipe M. G. França. Logical Reasoning via Satisfiability Mapped into Energy Functions. *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific, v. 22, p. 1031-1043, 2008.
10. Rodrigo R. Braga, Zhijun Yang, Felipe M. G. França. IMPLEMENTING AN ARTIFICIAL CENTIPEDE CPG: Integrating appendicular and axial movements of the scolopendromorph centipede. In: International Conference on Bio-inspired Systems and Signal Processing, 2008, Funchal. *Proceedings of BIOSIGNALS*. Setúbal : INSTICC – Press, 2008. v. 1. p. 58-62.

4.3 Orientações

São relatadas aqui, de forma não exaustiva, as atividades de formação, em andamento e concluídas, sob a forma de orientação exercida pelos pesquisadores do Projeto SCAE.

4.3.1 Orientações em andamento

Orientador: Alberto F. De Souza
Orientados: Bruno Zanetti Melotti (mestrado), Felipe Thomaz Pedroni (mestrado), Jairo Lucas de Moraes (mestrado), Nuno Rasseli (mestrado);

Orientador: Claudine Badue
Orientados: Caribe Zampirolli (mestrado), Rickson Guidolini (graduação),Vicente Bissoli (graduação);

Orientador: Elias Oliveira
Orientados: Lucas de Paula Veronese (graduação), Wallace Favoreto, (graduação), Marcia Goncalves Oliveira (mestrado), Patrick M. Ciarelli (doutorado);

Orientador: Felipe M. G. França
Orientados: Eliza França (graduação), Manoel V. M. França (graduação), Bruno F. Monteiro (mestrado), Lawrence C. Bandeira (mestrado), Bruno P. A. Grieco (doutorado);

Orientador: Priscila M. V. Lima
Orientados: Eliza França (graduação), Manoel V. M. França (graduação), Bruno F. Monteiro (mestrado), Bruno P. A. Grieco (doutorado).

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

4.3.2 Orientações concluídas

Orientador: Elias Oliveira
Orientado: Patrick M. Ciarelli (mestrado);

Orientador: Felipe M. G. França
Orientado: Ramon Diacovo (mestrado);

Orientador: Priscila M. V. Lima
Orientado: Ramon Diacovo (mestrado);

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

5 Participação da Equipe Científica em Encontros Relevantes

Ao longo do período foram realizados encontros e reuniões que contribuíram para o bom andamento do projeto.

5.1 Terceiro Encontro Técnico da Equipe SCAE

O Terceiro Encontro Técnico da Equipe SCAE, realizado em Vitória-ES de 14 a 16 de dezembro de 2007, contou com a participação das equipes de pesquisadores e alunos da UFES e UFRJ. No encontro foram debatidas questões científicas relacionadas ao aprimoramento dos modelos de categorização de texto no contexto do projeto. Foi feita uma verificação do andamento do cronograma de execução anteriormente definido e foram incluídas revisões e atualizações de tarefas previstas para a execução do projeto.

5.2 Workshop sobre incorporação de cores ao SCAE

Realizado em Vitória-ES, no dia 25 de março de 2008, esta oficina virtual (teleconferência) contou com a participação dos pesquisadores, técnicos e alunos da UFES e UFRJ. Uma apresentação técnica foi feita visando facilitar o entendimento das interfaces então estabelecidas no software, i.e., no sistema computacional em desenvolvimento, de forma a permitir a introdução de novos classificadores. Esta apresentação foi assistida pelas equipes da UFRJ e UFES.

5.3 Quarto Encontro Técnico da Equipe SCAE

O Quarto Encontro Técnico da Equipe SCAE, realizado em Vitória-ES de 11 a 12 de abril de 2008, contou com a participação das equipes de pesquisadores e alunos da UFES e UFRJ. Foram apresentadas a metodologia e o ambiente de desenvolvimento de Sistemas de Informação da Receita Federal para consideração no planejamento da infra-estrutura tecnológica dos projetos de pesquisa, em particular na do projeto SCAE.

5.4 Participação na XX Reunião Ordinária da Subcomissão Técnica para a CNAE-Subclasses

A XX Reunião Ordinária da Subcomissão Técnica para a CNAE -Subclasses, referenciada à Concla - Comissão Nacional de Classificação, ocorreu em Blumenau-SC, de 18 a 20 de junho de 2008. O evento contou com a participação da Pesquisadora Eliana Zadonade, que apresentou as considerações do Projeto SCAE quanto à definição dos requisitos para a Coleta Piloto. Um chamada pública para a referida Coleta foi definida no evento e um sítio web foi definido: http://www.scae.inf.br.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

5.5 Reunião de Definição de Tarefas e Responsabilidades na Coleta Piloto

Realizada na sede do IBGE, Rio de Janeiro-RJ, no dia 8 de julho de 2008, foi realizada uma reunião, cujo principal objetivo era a confecção de um Plano para Execução da Coleta Piloto, visando a definição das atuações, por parte de todos os envolvidos (Receita Federal, Universidades e IBGE). Os professores Alberto F. De Souza (UFES), Eliana Zadonade (UFES) e Felipe França (UFRJ), pesquisadores do Projeto SCAE, participaram desta reunião.

6 Lição Aprendida no Período

"*O grande objetivo da ciência é explicar o maior número de fatos empíricos com dedução lógica a partir do menor número de hipóteses e axiomas.*"— Albert Einstein.

Nas práticas e teorias sobre o bom funcionamento das organizações sociais, mesmo diante do estabelecimento de um planejamento adequado e antecipado, é sempre saudável manter uma avaliação contínua sobre os vários aspectos estratégicos para um sistema que preza o aprimoramento. Neste sentido, a intenção desta seção é identificar experiências e conhecimentos importantes adquiridos durante o período pertinente a este relato.

6.1 Quanto à importância dos dados

De maneira inquestionável, a importância dos dados se mostra definitiva, praticamente e independentemente de escopo:

"*Se a informação é a moeda corrente da nova economia, então os dados são a matéria prima crítica e necessária para o sucesso. Assim como uma refinaria transforma o oleo cru em inúmeros subprodutos do petróleo, empresas usam os dados para gerar uma multiplicidade de informações valiosas. Estas informações formam a base dos planos estratégicos e ações que determinarão o sucesso de uma firma. Consequentemente, dados de baixa qualidade podem ter um impacto negativo na saúde de uma empresa. Se não identificados e corrigidos a tempo, dados defeituosos podem contaminar todos os sistemas que os usam e o valor das informações obtidas.*" — Application Development Trends, Wayne W. Eckerson

Esta visao é coerente com a de outros pesquisadores na área de qualidade de dados:

"*Independentemente do fato de bancos de dados convencionais ou data warehaouses seram usados em apoio à tomada de decisões, está claro que o gerenciamento da qualidade das informações é critico para a efetividade dos sistemas de apoio à decisão empregados. No entanto, o gerenciamento da qualidade das informações pré-supõe um claro entendimento consensual com respeito ao significado do termo "qualidade da informação". De fato, questões fundamentais ainda persistem quanto a como qualidade deveria ser definida e quanto aos critérios específicos que deveriam ser adotados para avaliar a qualidade da informação. Definições de qualidade e os critérios de qualidade associados e suas categorias (usados para agrupar critérios) encontrados na prática e na literatura de sistemas de informação podem, em geral, ser descritos como advindos de perspectivas baseadas em produtos ou serviços, e empregando tanto abordagens empíricas, práticas, teóricas ou encontradas na literatura.*" — (2004) A Semiotic Information Quality Framework (DSS2004), Rosanne Price and Graeme Shanks.

Finalmente, uma última visão que coaduna com os nossos atuais esforços, principalmente no que tange à importância da execução da Coleta Piloto:

"*As pessoas freqüentemente assumem que qualidade de dados significa simplesmente a eliminação de dados ruins, dados em falta, imprecisos ou incorretos. Dados ruins certamente são um problema, mas não são o único problema. Um programa para uma boa qualidade de dados garante que os dados são compreensíveis, consistentes, relevantes e atuais.*" — Be Prepared to Duel with Data Quality, Rick Sherman.

Bibliografia

ALVARENGA, L. A. Teoria do Conceito Revisitada em Conexão com Ontologias e Metadados no Contexto das Bibliotecas Tradicionais e Digitais. DataGramaZero – Revista de Ciência da Informação, (2): 6, 2001.

ARAMPATZIS, A. T. et al. "Linguistically-motivated Information Retrieval". In: Encyclopedia of Library and Information Science. Nova York: Marcel Dekker, v. 69, 2000. p. 201-222.

ASPELL. GNU Aspell. Disponível: <http://aspell.net>. Acesso em: 20 de Agosto de 2008.

BAEZA-YATES, R.; RIBEIRO-NETO, B. Modern Information Retrieval. 1. ed. New York: Addison-Wesley, 1998.

COHEN, J. A. A coefficient of agreement for nominal scales. Educ. Psychol. Measmnt., (20), p. 37-46, 1960.

DALIANIS, H. Evaluating a Spelling Support in a Search Engine. In Proceedings of NLDB-2002, the 7th International Workshop on the Applications of Natural Language to Information Systems, June 2002.

DE CAMPOS, L. M., FERNÁNDEZ-LUNA, J. M., HUETE, J. F. The BNR Model: Foundations and Performance of a Bayesian Network-Based Retrieval Model. International Journal of Approximative Reasoning, (34), p. 265-285, 2003.

DIAS, M. A. L. Implementação do *Portuguese Stemmer* de ORENGO, V. M. e HUYCK, C. R. em Java. Disponível em: <http://ensino.univates.br/~mald/>. Acesso em: 11 de abril de 2008.

DNRC – DEPARTAMENTO NACIONAL DE REGISTRO DO COMÉRCIO. *Ranking* das Juntas Comerciais Segundo Movimento de Constituição, Alteração e Extinção e Cancelamento de Empresas. Ministério do Desenvolvimento, Indústria e Comércio Exterior – Secretaria do Desenvolvimento da Produção, Departamento Nacional de Registro do Comércio (DNRC), 2008.

FERNÁNDEZ-LUNA, J.M., Modelos de Recuperacíon Basados en Redes de Creencia. Ph.D. Thesis, Universidad de Granada, 2001.

FREUND, Y. Boosting a Weak Learning Algorithm by Majority. In: Proceedings of the Third Annual Workshop on Computational Learning Theory, p. 202-216, 1990.

FREUND, Y.; SCHAPIRE, R. E. A Decision-Theoretic Generalization of on-line Learning and an Application to Boosting. In: Proceedings of the Ninth Annual Conference on Computation Learning Theory, p. 325-332, 1996.

GIBBONS, A., Algorithmic Graph Theory. Cambridge University Press, Cambridge, UK, 1985.

GREGHI, J. G. Projeto e desenvolvimento de uma base de dados lexicais do português. Msc. Thesis. March, 2002.

GRISHMAN, R. Computational Linguistics: an Introduction. Cambridge Univ. Press, Cambridge, 1986.

HAYKIN, S. Neural Networks – A Comprehensive Foundation. 2. ed. New Jersey: Prentice Hall, 1998.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Classificação Nacional de Atividades Econômicas - CNAE 1.0 / CNAE-FISCAL 1.1. Technical report, Instituto Brasileiro de Geografia e Estatística (IBGE), 2003.

KEARNS, M. Thoughts on Hypothesis Boosting. Unpublished manuscript. Project for Ron Rivest's machine learning course at MIT. Disponível em: <http://www.cis.upenn.edu/~mkearns >. Acesso em: 15 de agosto de 2008.

KURAMOTO, H. Sintagmas Nominais: uma Nova Proposta para a Recuperação de Informação. DataGramaZero – Revista de Ciência da Informação, (3): 1, 2002.

LANCASTER, F. W. Indexação e Resumos: Teoria e Prática. 2. ed. Illinois: University of Illinois, 2003.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. Biometrics, (33), p. 159-174, 1977.

LIANG, S. The Java™ Native Interface: Programmer's Guide and Specification. Prentice Hall, 1999.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. An Introduction to Information Retrieval. Cambridge University Press, Cambridge, England, 2008.

MARTINS, D.; SILVA, M. J. Spelling Correction for Search Engine Queries. In Book Series of Lecture Notes in Computer Science, Vol. 3230, p. 372-383, 2004.

MIORELLI, S. T. ED-CER: Extração do Sintagma Nominal em Sentenças em Português. Dissertação de Mestrado, Programa de Pós Graduação em Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul. Porto Alegre, 2001.

MONTGOMERY, D. C. Design and Analysis of Experiments. John Willey and Sons, 2001.

MORETTIN, L. G. Estatística Básica, Volume II. 7 ed. Editora Makron Books, 1999.

NEAPOLITAN, R.E., Learning Bayesian Networks. Prentice Hall, Upper Saddle River, NJ, 2003.

OLIVEIRA, E. et. al. Classificando Automaticamente Documentos Digitais no Site de Notícias do UOL. In: XIV Seminário Nacional de Bibliotecas Universitárias. Salvador, 2006.

OLIVEIRA , E.; CIARELLI, P. M.; HENRIQUE, W. F.; VERONESE, L.; PEDRONI, F.; DE SOUZA, A. F. Intelligent Classification of Economic Activities from Free Text Desciptions. V Workshop em Tecnologia da Informação e da Linguagem Humana - TIL, 2007.

ORENGO, V. M.; HUYCK, C. R. A Stemming Algorithm for the Portuguese Language. In: Proceedings of the SPIRE Conference. Laguna de San Raphael: [s.n.], 2001, p. 13-15.

PARREIRAS, F. O Uso de Sintagmas Nominais como Fonte de Descritores para Textos de Periódicos Científicos. 2003. Disponível em: <http://www.fernando.parreiras.nom.br/>. Acesso em: 18 de agosto de 2008.

PEARL, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

PERINI, M. A. et al. O SN em Português: A Hipótese Mórfica. Revista de Estudos de Linguagem – UFMG, Belo Horizonte, p. 43–56, 1996.

PIEDADE, M. A. R. Introdução á Teoria da Classificação. 2. ed. Rio de Janeiro: Interciência, 1977.

PORTER, M. F. An Algorithm for Suffix Stripping. Program, (14): 3. p. 130-137, 1980.

REICHENBACH, H. Elements of symbolic logic. Berkeley, CA: University of California Press. 1947.

SCHAPIRE, R. E. The strength of Weak Learnability. Machine Learning, 5(2): 197-227, June, 1990.

SCHAPIRE, R. E.; SINGER, Y. Improved boosting algorithms using confidence-rated predictions. Machine Learning, 27(3):297–336, 1999.

SCHAPIRE, R. E.; SINGER, Y. BoosTexter: A Boosting-based System for Text Categorization. Machine Learning, 2000, ed. 39, p. 135-168.

SEBASTIANI, F. Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1): p. 1-47, 2002.

SHIMAKURA, S. E. Interpretação do coeficiente de correlação. CE003 Estatística II, Notas de Aula. Laboratório de Estatística e Geoinformação – UFPR. Disponível em: <http://leg.ufpr.br/~silvia/CE003/node74.html>. Acesso em: 18 de agosto de 2008.

SIEGEL, S.; CASTELLAN, N. Nonparametric Statistics for the Behavioral Sciences. 2 ed. New York: McGraw-Hill, 1988. 284-285.

SOARES, J. F.; SIQUEIRA, A. L. Introdução a Estatística Médica. Departamento de Estatística / UFMG, 1999.

SOUCY, P.; MINEAU, G. W. A Simple KNN Algorithm for Text Categorization. In: ICDM'01: Proceedings of the 2001 IEEE International Conference on Data Mining. Washington, DC, USA: IEEE Computer Society. p. 647–648, 2001.

SOUZA, R. R. Uma Proposta de Metodologia para Escolha Automática de Descritores Utilizando Sintagmas Nominais. Tese (Doutorado) – Departamento de Ciências da Informação, UFMG, Belo Horizonte, 2005.

TRIOLA, M. F. Introdução a Estatística. Editora LTC – Livros Técnicos e Científicos Editora S. A., 1999.

VIOLA, P.; JONES, M. Rapid Object Detection using a Boosted Cascade of Simple Features. In: Proceedings of Computer Vision and Pattern Recognition, 2001.

ZHANG, M.-L.; ZHOU, Z.-H. ML-KNN: A Lazy Learning Approach to Multi-Label Learning. Pattern Recognition, 40(7): p. 2038-2048, 2007.

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

ANEXO 1: Distribuição de Freqüências por Subclasse – Base de Vitória + BH

Subclasse	Frequência	%
7040800	8612	4,53
7416002	5600	2,95
5232900	4632	2,44
7420902	4171	2,19
5119500	3261	1,72
5249399	2956	1,56
4521701	2954	1,55
5522000	2879	1,51
8099305	2272	1,20
5249303	2126	1,12
5118700	2102	1,11
5020201	2049	1,08
5030003	2005	1,05
5245002	1984	1,04
5521201	1810	0,95
5246902	1785	0,94
5231002	1771	0,93
7010600	1649	0,87
7250800	1613	0,85
5213202	1605	0,84
5242601	1576	0,83
7499399	1534	0,81
7290700	1500	0,79
9199500	1442	0,76
5241804	1407	0,74
5233701	1402	0,74
7229000	1389	0,73
5229999	1384	0,73
7221400	1356	0,71
7440301	1274	0,67
5521202	1270	0,67
8513801	1266	0,67
7230300	1206	0,63
6720201	1197	0,63
5243401	1186	0,62
6599403	1186	0,62
5224800	1159	0,61
7210900	1148	0,60
4541101	1110	0,58
7499307	1069	0,56
7032700	1065	0,56
7499302	1058	0,56
7412801	1057	0,56
7020300	998	0,53
5244201	980	0,52
7499308	951	0,50
2996399	918	0,48
5243499	917	0,48
6026701	910	0,48
6026702	908	0,48
5244208	903	0,48
5241801	888	0,47

Subclasse	Frequência	%
7411001	877	0,46
5271001	875	0,46
7499312	874	0,46
5249306	840	0,44
7110200	798	0,42
5246901	794	0,42
5221301	792	0,42
7450002	792	0,42
8014400	792	0,42
9231203	783	0,41
1812001	782	0,41
5233702	778	0,41
7440399	772	0,41
7139099	717	0,38
5245003	693	0,36
6330400	691	0,36
7420901	674	0,35
5249302	672	0,35
8513802	664	0,35
9302501	663	0,35
6321503	656	0,35
7031900	646	0,34
5192600	644	0,34
5231003	637	0,34
5142001	629	0,33
5020203	627	0,33
9302502	624	0,33
2229202	620	0,33
2229299	618	0,33
5222100	611	0,32
5223000	605	0,32
6720299	601	0,32
7470501	600	0,32
5244205	586	0,31
5249301	568	0,30
5010506	564	0,30
5221302	552	0,29
9191000	534	0,28
7132300	525	0,28
5249305	518	0,27
5139099	500	0,26
5010502	499	0,26
8099303	489	0,26
4513600	481	0,25
5169199	470	0,25
5279599	468	0,25
7133100	459	0,24
5244299	454	0,24
5249307	448	0,24
5010507	447	0,24
4550099	445	0,23
5245001	445	0,23

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
5246903	445	0,23
5050400	440	0,23
4521702	430	0,23
7420999	430	0,23
5242602	428	0,23
5229902	427	0,22
7420905	420	0,22
7450001	419	0,22
4543801	418	0,22
5149799	418	0,22
5241805	418	0,22
7240000	415	0,22
4522501	411	0,22
7140403	407	0,21
5149701	405	0,21
5231001	405	0,21
8099304	399	0,21
2229201	393	0,21
6412202	390	0,21
5020205	384	0,20
5145403	382	0,20
6599407	377	0,20
5030001	374	0,20
5242604	372	0,20
7412802	372	0,20
5030006	359	0,19
5271002	358	0,19
5249311	352	0,19
7491801	352	0,19
2219500	346	0,18
8099399	345	0,18
5116000	341	0,18
2839800	339	0,18
5165901	339	0,18
5169101	339	0,18
2215200	338	0,18
6522600	338	0,18
2222502	334	0,18
6559503	333	0,18
7414400	332	0,17
7499305	332	0,17
5117900	330	0,17
5247700	326	0,17
6719999	325	0,17
8015200	324	0,17
4529203	318	0,17
6412201	318	0,17
4550004	317	0,17
7460802	315	0,17
5113600	309	0,16
7491803	306	0,16
2992099	303	0,16

Subclasse	Frequência	%
5169102	298	0,16
5146201	292	0,15
6521800	292	0,15
9262202	287	0,15
8515499	285	0,15
5147001	281	0,15
8515403	278	0,15
7413600	272	0,14
1581402	268	0,14
8515404	268	0,14
5114400	267	0,14
5279503	265	0,14
8099301	265	0,14
7499306	264	0,14
5243403	258	0,14
5136599	255	0,13
5244202	255	0,13
9261404	254	0,13
9261405	252	0,13
5279501	251	0,13
4533002	249	0,13
5153599	247	0,13
9304100	245	0,13
7310500	244	0,13
8013600	241	0,13
4550001	236	0,12
6719906	232	0,12
7420903	229	0,12
4533001	227	0,12
4550005	227	0,12
5241803	224	0,12
6340103	224	0,12
4523300	222	0,12
6023201	220	0,12
9000001	219	0,12
0161901	217	0,11
5244206	216	0,11
5153503	214	0,11
5279502	213	0,11
5524702	213	0,11
9222302	213	0,11
5145401	211	0,11
5214000	211	0,11
5241806	210	0,11
5191801	209	0,11
5244203	204	0,11
5030004	202	0,11
5249312	201	0,11
7139004	196	0,10
4529299	195	0,10
6420380	193	0,10
9120000	193	0,10

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
9211802	193	0,10
5243402	191	0,10
9301701	189	0,10
2216000	186	0,10
3391000	185	0,10
3611001	182	0,10
5144601	180	0,09
5244204	180	0,09
5519099	180	0,09
4525001	179	0,09
5144602	178	0,09
7470502	178	0,09
5145405	177	0,09
5212400	174	0,09
8516299	171	0,09
5147002	170	0,09
9262206	170	0,09
5020202	169	0,09
5249315	169	0,09
6599499	169	0,09
5154399	168	0,09
5513101	168	0,09
2221700	167	0,09
5249310	167	0,09
5229901	166	0,09
8513899	166	0,09
8514604	166	0,09
7440302	165	0,09
4542000	164	0,09
5215902	164	0,09
1811201	163	0,09
8511100	162	0,09
5213201	160	0,08
7140499	160	0,08
7499301	160	0,08
4550006	159	0,08
8514699	159	0,08
8514602	158	0,08
5243404	155	0,08
9261402	155	0,08
5141104	153	0,08
5020204	152	0,08
5115200	151	0,08
5159401	151	0,08
2842800	150	0,08
7511600	149	0,08
4512801	148	0,08
1812002	147	0,08
5143800	147	0,08
8020900	147	0,08
2217900	145	0,08
9211899	145	0,08

Subclasse	Frequência	%
1581401	144	0,08
7140401	144	0,08
5030005	140	0,07
5141102	140	0,07
3699499	139	0,07
8520000	138	0,07
9309202	138	0,07
5149703	137	0,07
5164001	137	0,07
6024002	137	0,07
7430600	137	0,07
7499303	137	0,07
5153505	136	0,07
5249304	136	0,07
6340104	136	0,07
6593501	135	0,07
3611002	134	0,07
5524701	131	0,07
8099307	131	0,07
5146202	129	0,07
9111100	129	0,07
6025906	128	0,07
5242603	127	0,07
8099306	127	0,07
5041504	126	0,07
9112000	126	0,07
5149708	125	0,07
5161600	124	0,07
2969600	123	0,06
6611701	122	0,06
5250799	121	0,06
2812600	120	0,06
5513103	119	0,06
6340101	119	0,06
7320200	119	0,06
3189500	118	0,06
4522503	117	0,06
8031400	117	0,06
5241802	115	0,06
1589099	114	0,06
6025903	111	0,06
5152700	110	0,06
6340199	110	0,06
2218700	109	0,06
6024003	108	0,06
3310305	107	0,06
4550003	107	0,06
2899100	105	0,06
5010501	104	0,05
6420321	104	0,05
6420311	103	0,05
6025902	100	0,05

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
5041503	99	0,05
7140402	99	0,05
6630300	98	0,05
4541102	95	0,05
5153506	95	0,05
7420904	95	0,05
8515405	95	0,05
5112800	94	0,05
9231202	93	0,05
9231299	92	0,05
3691902	91	0,05
5165902	91	0,05
5042300	90	0,05
6559501	90	0,05
3450900	87	0,05
8532499	85	0,04
9239803	85	0,04
6720203	84	0,04
6024001	83	0,04
6026703	83	0,04
5155101	82	0,04
5524703	82	0,04
6720202	82	0,04
8514601	82	0,04
9261403	82	0,04
1584900	79	0,04
4549799	79	0,04
5141103	79	0,04
9261401	79	0,04
5164002	78	0,04
5145402	76	0,04
4525003	75	0,04
8032200	75	0,04
2994700	74	0,04
4550002	74	0,04
6312602	74	0,04
3393600	73	0,04
8096900	73	0,04
9240100	73	0,04
5142002	72	0,04
6028301	72	0,04
1931301	71	0,04
5191802	71	0,04
4560800	70	0,04
5020206	70	0,04
5249308	70	0,04
7416001	70	0,04
9211804	70	0,04
5111000	68	0,04
2214400	67	0,04
6025905	67	0,04
9239804	67	0,04

Subclasse	Frequência	%
5279504	66	0,03
5529800	66	0,03
7121800	66	0,03
2992003	65	0,03
4531402	65	0,03
5139007	65	0,03
6311800	65	0,03
6523400	65	0,03
1811202	64	0,03
6411401	63	0,03
9232004	63	0,03
2691303	62	0,03
5131400	62	0,03
5139009	62	0,03
5142003	62	0,03
5211600	62	0,03
4543802	61	0,03
1813901	60	0,03
5229903	60	0,03
7139003	60	0,03
7491805	60	0,03
9221500	60	0,03
2229203	59	0,03
6612501	59	0,03
4512802	58	0,03
6612599	58	0,03
5153504	57	0,03
7499313	57	0,03
4511001	56	0,03
5030002	56	0,03
6524202	56	0,03
8033000	56	0,03
8515401	56	0,03
9251700	56	0,03
5139004	55	0,03
5154301	55	0,03
9309299	55	0,03
3199200	54	0,03
4011800	54	0,03
4531403	54	0,03
5149706	54	0,03
5134900	53	0,03
5153501	53	0,03
5159499	53	0,03
7139001	53	0,03
9262201	53	0,03
4549704	51	0,03
5136502	51	0,03
6321599	51	0,03
9213400	51	0,03
9303304	51	0,03
3699401	49	0,03

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
2473200	48	0,03
5136501	48	0,03
1821000	47	0,02
5159403	47	0,02
2929700	46	0,02
3290501	46	0,02
5133001	46	0,02
5136503	46	0,02
5139008	46	0,02
5169103	46	0,02
6559502	46	0,02
8531699	45	0,02
3310303	44	0,02
5215901	44	0,02
9000099	44	0,02
1543100	43	0,02
5513102	43	0,02
6420399	43	0,02
9261499	43	0,02
1750701	42	0,02
2991202	42	0,02
5121707	42	0,02
5139005	42	0,02
6023202	42	0,02
6411402	42	0,02
1761200	41	0,02
5250701	41	0,02
6622200	41	0,02
8516201	41	0,02
9212600	41	0,02
1929100	40	0,02
2222503	40	0,02
2630101	40	0,02
3290502	40	0,02
3612902	40	0,02
4511002	40	0,02
2029001	39	0,02
4012600	39	0,02
5272800	39	0,02
0141401	38	0,02
2529199	38	0,02
6535800	38	0,02
9231201	38	0,02
9239899	38	0,02
3181003	37	0,02
3691901	37	0,02
5149704	37	0,02
8531601	37	0,02
1582200	36	0,02
1921600	36	0,02
2893200	35	0,02
3613702	35	0,02

Subclasse	Frequência	%
4522502	35	0,02
5149702	35	0,02
5153507	35	0,02
5519005	35	0,02
6323199	35	0,02
1429099	34	0,02
2472400	34	0,02
2811800	34	0,02
9262299	34	0,02
5145404	33	0,02
7514000	33	0,02
4014200	32	0,02
1750799	31	0,02
2519400	31	0,02
5135700	31	0,02
5141101	31	0,02
5151901	31	0,02
6420392	31	0,02
3330800	30	0,02
4549703	30	0,02
5121701	30	0,02
5137302	30	0,02
6210300	30	0,02
7411003	30	0,02
7460804	30	0,02
9211801	30	0,02
2022202	29	0,02
4529205	29	0,02
5041502	29	0,02
6025904	29	0,02
6220002	29	0,02
6322301	29	0,02
7524800	29	0,02
1310201	28	0,01
1741800	28	0,01
2232200	28	0,01
2499600	28	0,01
2630199	28	0,01
2995502	28	0,01
3532700	28	0,01
4529201	28	0,01
6340102	28	0,01
8515402	28	0,01
1585700	27	0,01
1939900	27	0,01
2996301	27	0,01
4100900	27	0,01
5121799	27	0,01
5523901	27	0,01
6621400	27	0,01
0161999	26	0,01
2022299	26	0,01

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
3310302	26	0,01
3612901	26	0,01
4549701	26	0,01
5151903	26	0,01
6025901	26	0,01
7131500	26	0,01
7491806	26	0,01
2454600	25	0,01
3340504	25	0,01
4529202	25	0,01
7492600	25	0,01
9262208	25	0,01
0213500	24	0,01
2029002	24	0,01
2892401	24	0,01
5132201	24	0,01
6312601	24	0,01
8513803	24	0,01
1769800	23	0,01
2940800	23	0,01
4525002	23	0,01
5121703	23	0,01
5155102	23	0,01
6024004	23	0,01
6593502	23	0,01
9222301	23	0,01
0212701	22	0,01
1324201	22	0,01
2991204	22	0,01
3152600	22	0,01
4531401	22	0,01
5149707	22	0,01
6220001	22	0,01
6719904	22	0,01
2699900	21	0,01
2995504	21	0,01
3210700	21	0,01
3310301	21	0,01
5041501	21	0,01
5041505	21	0,01
5122506	21	0,01
5151906	21	0,01
5153502	21	0,01
6027500	21	0,01
6712105	21	0,01
9232001	21	0,01
1410903	20	0,01
2231400	20	0,01
2691302	20	0,01
2993901	20	0,01
3694399	20	0,01
3710999	20	0,01

Subclasse	Frequência	%
5122505	20	0,01
6712101	20	0,01
1410902	19	0,01
2522400	19	0,01
2993902	19	0,01
3340503	19	0,01
4013400	19	0,01
5122501	19	0,01
5139001	19	0,01
6322399	19	0,01
9231204	19	0,01
1410999	18	0,01
1513001	18	0,01
1813902	18	0,01
2952100	18	0,01
2991201	18	0,01
4529204	18	0,01
5151904	18	0,01
5159402	18	0,01
5244207	18	0,01
6323102	18	0,01
6712102	18	0,01
9262207	18	0,01
2234900	17	0,01
2512700	17	0,01
5132202	17	0,01
5139003	17	0,01
6420391	17	0,01
7499310	17	0,01
8516202	17	0,01
0141402	16	0,01
2141500	16	0,01
2452001	16	0,01
2995503	16	0,01
5010503	16	0,01
5139006	16	0,01
5149705	16	0,01
6592700	16	0,01
6613300	16	0,01
7460801	16	0,01
1511301	15	0,01
1822800	15	0,01
2149099	15	0,01
2529103	15	0,01
2751000	15	0,01
2924600	15	0,01
6712104	15	0,01
6712106	15	0,01
7140405	15	0,01
8512000	15	0,01
0161903	14	0,01
1542300	14	0,01

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
1586500	14	0,01
2630102	14	0,01
2892499	14	0,01
2989000	14	0,01
3021000	14	0,01
3121600	14	0,01
3613701	14	0,01
5121709	14	0,01
5155103	14	0,01
6420319	14	0,01
7411004	14	0,01
2131800	13	0,01
2222501	13	0,01
3392800	13	0,01
3449502	13	0,01
3695100	13	0,01
5523902	13	0,01
6010002	13	0,01
6420329	13	0,01
7140404	13	0,01
9000002	13	0,01
9000003	13	0,01
9211803	13	0,01
9239801	13	0,01
9500100	13	0,01
0122800	12	0,01
1764700	12	0,01
2481300	12	0,01
2630105	12	0,01
3022800	12	0,01
3693500	12	0,01
5133002	12	0,01
5154302	12	0,01
6028302	12	0,01
6540400	12	0,01
6551000	12	0,01
6712103	12	0,01
8097700	12	0,01
1512101	11	0,01
1762000	11	0,01
2529101	11	0,01
2619000	11	0,01
2713800	11	0,01
2923800	11	0,01
2992004	11	0,01
5122504	11	0,01
6420330	11	0,01
7139002	11	0,01
7512400	11	0,01
8516207	11	0,01
9252502	11	0,01
9262204	11	0,01

Subclasse	Frequência	%
1429001	10	0,01
1583001	10	0,01
1749300	10	0,01
1779500	10	0,01
2121000	10	0,01
2833900	10	0,01
2961000	10	0,01
2991205	10	0,01
3350200	10	0,01
3697800	10	0,01
7411002	10	0,01
7491804	10	0,01
8531603	10	0,01
0142202	9	0,00
0146599	9	0,00
1310202	9	0,00
1329304	9	0,00
1750702	9	0,00
2022201	9	0,00
2749999	9	0,00
2843600	9	0,00
3192500	9	0,00
3439800	9	0,00
3599800	9	0,00
5132203	9	0,00
5139002	9	0,00
5249314	9	0,00
6112300	9	0,00
6321501	9	0,00
8099302	9	0,00
8514606	9	0,00
9303399	9	0,00
0144900	8	0,00
0162799	8	0,00
1329305	8	0,00
1511302	8	0,00
1559800	8	0,00
1931302	8	0,00
2451100	8	0,00
2471600	8	0,00
2641701	8	0,00
2642500	8	0,00
2649299	8	0,00
2726002	8	0,00
3012000	8	0,00
3181002	8	0,00
5151905	8	0,00
5215903	8	0,00
5249309	8	0,00
6559599	8	0,00
7499311	8	0,00
7513200	8	0,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
9303301	8	0,00
1120700	7	0,00
1410906	7	0,00
1556300	7	0,00
2142300	7	0,00
2321300	7	0,00
2419800	7	0,00
2813400	7	0,00
2921100	7	0,00
2931900	7	0,00
2996302	7	0,00
3320000	7	0,00
3340502	7	0,00
3720600	7	0,00
5122502	7	0,00
5519001	7	0,00
6025907	7	0,00
6111500	7	0,00
6321504	7	0,00
6420312	7	0,00
6599402	7	0,00
6599405	7	0,00
7123400	7	0,00
9253300	7	0,00
0145703	6	0,00
1110001	6	0,00
1324202	6	0,00
1523700	6	0,00
1571702	6	0,00
1589004	6	0,00
2132600	6	0,00
2453800	6	0,00
2691301	6	0,00
2714600	6	0,00
2915700	6	0,00
2996304	6	0,00
3112700	6	0,00
3122400	6	0,00
3142902	6	0,00
3394400	6	0,00
3431200	6	0,00
3441000	6	0,00
3523800	6	0,00
3592000	6	0,00
3692700	6	0,00
5121702	6	0,00
5154303	6	0,00
6322303	6	0,00
6420340	6	0,00
6420352	6	0,00
6534001	6	0,00
6611702	6	0,00

Subclasse	Frequência	%
9192800	6	0,00
0139299	5	0,00
0512601	5	0,00
1421400	5	0,00
1571701	5	0,00
1583002	5	0,00
1731000	5	0,00
1771000	5	0,00
1933000	5	0,00
2010901	5	0,00
2491000	5	0,00
2521600	5	0,00
2529102	5	0,00
2725199	5	0,00
2741301	5	0,00
2752900	5	0,00
2912200	5	0,00
2992001	5	0,00
2992005	5	0,00
3511401	5	0,00
3512202	5	0,00
3614500	5	0,00
3710901	5	0,00
4549702	5	0,00
5121704	5	0,00
5133003	5	0,00
5249313	5	0,00
6321502	5	0,00
7122600	5	0,00
7499304	5	0,00
7499309	5	0,00
7521300	5	0,00
7525600	5	0,00
8532402	5	0,00
9301702	5	0,00
9303303	5	0,00
0161902	4	0,00
0161904	4	0,00
1410904	4	0,00
1513002	4	0,00
1514800	4	0,00
1521000	4	0,00
1522900	4	0,00
1591102	4	0,00
1595402	4	0,00
2021400	4	0,00
2149001	4	0,00
2429599	4	0,00
2431700	4	0,00
2494500	4	0,00
2611500	4	0,00
2630103	4	0,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
2914900	4	0,00
2991203	4	0,00
2995501	4	0,00
2996303	4	0,00
3142901	4	0,00
3221200	4	0,00
3222000	4	0,00
3531900	4	0,00
4020702	4	0,00
5137301	4	0,00
6323101	4	0,00
6420322	4	0,00
6531500	4	0,00
6532300	4	0,00
6720204	4	0,00
8514603	4	0,00
8515406	4	0,00
9262205	4	0,00
0112000	3	0,00
0119899	3	0,00
0121099	3	0,00
0132500	3	0,00
0145704	3	0,00
0146501	3	0,00
0146503	3	0,00
0211901	3	0,00
0211905	3	0,00
0512605	3	0,00
1000602	3	0,00
1323401	3	0,00
1591101	3	0,00
1593802	3	0,00
1595401	3	0,00
2483000	3	0,00
2492901	3	0,00
2496100	3	0,00
2641702	3	0,00
2724301	3	0,00
2731600	3	0,00
2739100	3	0,00
2821500	3	0,00
2831200	3	0,00
2832000	3	0,00
2881900	3	0,00
2882700	3	0,00
2954800	3	0,00
2962900	3	0,00
2992002	3	0,00
3160700	3	0,00
3181001	3	0,00
3230100	3	0,00
3410001	3	0,00

Subclasse	Frequência	%
3511403	3	0,00
5010504	3	0,00
5121706	3	0,00
5122503	3	0,00
5151902	3	0,00
6122001	3	0,00
6559505	3	0,00
6559506	3	0,00
8514605	3	0,00
8531602	3	0,00
9232002	3	0,00
0111202	2	0,00
0111299	2	0,00
0139201	2	0,00
0143000	2	0,00
0145701	2	0,00
0162701	2	0,00
1000601	2	0,00
1323402	2	0,00
1410907	2	0,00
1429003	2	0,00
1531800	2	0,00
1589002	2	0,00
1594600	2	0,00
1721300	2	0,00
1732900	2	0,00
1733700	2	0,00
2010902	2	0,00
2023000	2	0,00
2110500	2	0,00
2122900	2	0,00
2429501	2	0,00
2432500	2	0,00
2452002	2	0,00
2461900	2	0,00
2482100	2	0,00
2495300	2	0,00
2511900	2	0,00
2723500	2	0,00
2726001	2	0,00
2741302	2	0,00
2834700	2	0,00
2913000	2	0,00
2996305	2	0,00
3182800	2	0,00
3432000	2	0,00
3443600	2	0,00
3444400	2	0,00
3512201	2	0,00
3691903	2	0,00
3694302	2	0,00
4030400	2	0,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
5010505	2	0,00
5121705	2	0,00
5121708	2	0,00
5519002	2	0,00
6021600	2	0,00
6123901	2	0,00
6322302	2	0,00
6420351	2	0,00
6524203	2	0,00
6533100	2	0,00
6591901	2	0,00
6719901	2	0,00
6719905	2	0,00
7530200	2	0,00
8531604	2	0,00
9309201	2	0,00
9309203	2	0,00
0111201	1	0,00
0111203	1	0,00
0119801	1	0,00
0119802	1	0,00
0119803	1	0,00
0119805	1	0,00
0119806	1	0,00
0119809	1	0,00
0119810	1	0,00
0119816	1	0,00
0119817	1	0,00
0121001	1	0,00
0121002	1	0,00
0121003	1	0,00
0131701	1	0,00
0131799	1	0,00
0134100	1	0,00
0139202	1	0,00
0139203	1	0,00
0139206	1	0,00
0139207	1	0,00
0139208	1	0,00
0139209	1	0,00
0139211	1	0,00
0139212	1	0,00
0145702	1	0,00
0146506	1	0,00
0161905	1	0,00
0211906	1	0,00
0512604	1	0,00
0512606	1	0,00
1321801	1	0,00
1321802	1	0,00
1322602	1	0,00
1329301	1	0,00

Subclasse	Frequência	%
1329303	1	0,00
1410901	1	0,00
1410908	1	0,00
1410909	1	0,00
1410910	1	0,00
1552000	1	0,00
1553900	1	0,00
1572500	1	0,00
1589003	1	0,00
1589005	1	0,00
1719100	1	0,00
1722100	1	0,00
1763900	1	0,00
1772800	1	0,00
1910000	1	0,00
1932100	1	0,00
2329901	1	0,00
2412000	1	0,00
2413900	1	0,00
2422800	1	0,00
2441400	1	0,00
2469400	1	0,00
2620400	1	0,00
2692100	1	0,00
2725101	1	0,00
2742100	1	0,00
2749902	1	0,00
2841000	1	0,00
2891600	1	0,00
2911400	1	0,00
2922000	1	0,00
2925400	1	0,00
2964500	1	0,00
2965300	1	0,00
2971800	1	0,00
3011200	1	0,00
3113500	1	0,00
3151800	1	0,00
3340501	1	0,00
3442800	1	0,00
3449501	1	0,00
3511402	1	0,00
3521100	1	0,00
3522000	1	0,00
4020701	1	0,00
5269800	1	0,00
6022400	1	0,00
6029100	1	0,00
6030500	1	0,00
6122002	1	0,00
6123902	1	0,00
6510200	1	0,00

Fundação Espírito-Santense de Tecnologia
Av. Fernando Ferrari, 845 | Goiabeiras | 29.060-410 - Vitória - ES
Tel: +55 27 3345-7555 | Fax: +55 27 3345-7668 | www.fest.org.br

Subclasse	Frequência	%
6534002	1	0,00
6534003	1	0,00
6591902	1	0,00
6599401	1	0,00
6711301	1	0,00
6711303	1	0,00
6711304	1	0,00
7460803	1	0,00
7522100	1	0,00
7523000	1	0,00
8516205	1	0,00
9239802	1	0,00
9252501	1	0,00
9262203	1	0,00
9303305	1	0,00
Total	190064	100,00