UNILASALLE

CENTRO UNIVERSITÁRIO LA SALLE

Curso de Bacharelado em Ciência da Computação

Adriano Ferreira Silveira

MODELO PARA DETECÇÃO DE REDES SOCIAIS ATRAVÉS DE ANÁLISE DE SIMILARIDADE EM TEXTOS DE BLOGS

Canoas, novembro de 2009

ADRIANO FERREIRA SILVEIRA

MODELO PARA DETECÇÃO DE REDES SOCIAIS ATRAVÉS DE ANÁLISE DE SIMILARIDADE EM TEXTOS DE BLOGS

Trabalho de conclusão apresentado em sessão de apresentação pública do curso de Ciência da Computação do Centro Universitário La Salle, como exigência parcial para a obtenção do grau de Bacharel em Ciência da Computação, sob orientação da Profª. DSc. Patrícia Kayser Vargas Mangan.

Canoas, novembro de 2009

AGRADECIMENTOS

Agradeço a toda minha família,
A minha mulher que tem sido uma boa companheira,
Aos meus filhos e enteados por compreenderem,
A minha mãe por sempre ter acreditado em mim,
A minha irmã por ser uma segunda mãe,
Ao meu pai e ao irmão por serem bons exemplos,
E a todos por existirem.

RESUMO

Este trabalho propõe um estudo sobre análise de similaridade entre blogs, baseando-se no conteúdo de suas postagens, e interação de seus usuários, buscando avaliar a intensidade de suas relações, assim como elucidar conexões implícitas, utilizando técnicas de mineração de dados, e posteriormente avaliar as similaridades encontradas.

Utilizando as técnicas de regra do cosseno, mineração de dados utilizando os algoritmos de clusterização EM (*Expectation Maximization*) e K-Means, com o auxílio das ferramentas WEKA e MathWorks Matlab, foi possível encontrar similaridade entre postagens, agrupá-los e através de técnicas de avaliação de *clusters*, avaliar a coesão dos agrupamentos encontrados.

PALAVRAS-CHAVE: *Text Mining*, Cibercultura, Redes Sociais

ABSTRACT

This article proposes a study on similarity between weblogs, based on the content of their posts, and interaction of its users, seeking to assess the intensity of their relationships, as well as elucidating implicit connections, using techniques of data mining, and further evaluate the similarities found

Using the techniques of the cosine rule, data mining algorithms using clustering EM (Expectation Maximization) and K-Means, with the help of WEKA and MathWorks Matlab tools, we found a similarity between threads, and group them using techniques assessment of clusters, assess the cohesion of groups found.

KEYWORDS: *Text Mining, Cyberculture, Social Networks*

LISTA DE ABREVIATURAS E SIGLAS

AGNES	*Agglomerative Nesting*
CA	Coeficiente Aglomerativo
CD	Coeficiente Divisivo
CVAP	*Cluster Validity Analysis Platform*
DIANA	*Divisive Analysis*
EM	*Expectation Maximization*
KDD	*Knowledge Discovery in Database*
KDT	*Knowlegde Discovery in Text*
MLE	Avaliação de Probabilidade Máxima
MONA	*Monothetic Analysis*
OO	Orientação a Objetos
RIA	*Rich Internet Application*
RSLP	Removedor de Sufixos da Língua Portuguesa
TF-IDF	*Term-Frequency Inverse-Document-Frequency*
UML	*Unified Modeling Language*
WEKA	*Waikato Environment for Knowledge Analysis*
WPF	Windows Presentation Foundation
XAML	*Extensible Application Markup Language*
XML	*Extensible Markup Language*

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

1.1 Contexto

O acesso à informação é a base para o desenvolvimento de indivíduos, grupos, comunidades e a sociedade em todas as suas áreas. Desde a criação da Internet e das redes sociais através desta, barreiras têm sido transpostas, relacionamentos fisicamente improváveis se tornaram corriqueiros e por conseqüência, o volume de informações contidas e trafegadas na rede aumentou. Neste cenário, as pessoas têm acesso a informações e também assumem o papel de autores. O intuito de compreender melhor este contexto é desenvolvido estudos de diversas áreas acadêmicas.

Uma rede social é composta por atores e suas conexões. Os atores podem ser representados por indivíduos, instituições ou grupos, e por sua vez as conexões serem classificadas quanto à sua direção (bidirecional ou unidirecional) e quanto a sua intensidade. O fruto da interação entre estes entes sociais através das conexões estabelecidas é o conhecido como Capital Social (RECUERO, 2005). O Capital Social de uma rede social, segundo (RECUERO, 2005), é o conjunto de recursos obtidos através da união e a interação dos recursos individuais, tornando-os recursos comuns a todos.

O fenômeno dos *blogs* [1] obteve uma maior visibilidade no Brasil a partir do ano 2000, embora em outros países os blogs já fossem realidade. O primeiro blog que se tem notícia surgiu em meados dos anos 90, com o propósito de ser um painel de notícias, o "*What`s new in 92*", nome do suposto primeiro *blog*, tinha apenas a proposta de ser um painel de informações sobre o projeto *World Wide Web* (MENDES, 2008).

Nos *blogs*, como são popularmente conhecidos os *weblogs*, as interações acontecem através da postagem de textos e comentários, lista de *blogs* recomendados (*blogroll*) e o uso de *trackbacks* (*links* que relacionam outros *blogs* que estão discutindo o mesmo assunto, ou é um dos interesses do proprietário do *blog*), criando redes sociais denominadas *webrings*, que

[1] *Website* extremamente flexibilizado com mensagens organizadas em ordem cronológica reversa e com uma interface de edição simplificada (Silva, 2003)

são redes hipertextuais e complexas, e que por sua vez podem gerar novas comunidades virtuais (PRIMO, 2003).

1.2 Problema de Pesquisa

O cenário descrito acima exibe uma estrutura de rede social, representada por *blogs*, autores (blogueiros) e colaboradores, bem como funcionam suas conexões. Neste cenário todas as conexões são feitas explicitamente, desde o comentário, que necessita da interação do comentarista, até o *trackback* que é inserido em uma discussão manualmente, onde é necessário um prévio conhecimento de outros blogs, onde discussões similares são abordadas. Apesar da utilização de *tags* para categorização de *blogs* e *posts*, em algumas situações, o conteúdo abordado pode ser muito amplo, podendo assim a categoria ser segmentada em subcategorias mais específicas. Assim pode um *blog* estar em uma categoria, mas abordar assuntos de mais de uma categoria, ou ainda pior, estar em uma categoria e tratar de assunto de outra totalmente diferente.

Assim, neste cenário, é desejável criar mecanismos de agrupamento de blogs, procurando elucidar conexões que estão implícitas no universo das redes sociais criadas a partir de blogs. Atualmente, os links são feitos manualmente através de *trackbacks*, a criação de *tags* (categorias criadas pelo blogueiro) para segmentar e organizar os assuntos de seu interesse pode auxiliar no momento de uma busca através de motores de busca, mas por sua vez podem ser únicas, não sendo suficientes no momento das buscas, ou até ser idêntica a mesma *tag* de outro *blog*, mas que trata de assunto completamente diferente.

Sem a elucidação de conexões implícitas, associado ao grande número de blogs que tratam dos mais diversos assuntos, há uma grande pulverização de conhecimento. Evitam-se, deste modo, que redes similares que poderiam se tornar em apenas uma grande rede, com mais autores, comentaristas e *trackbacks*, tornem-se um grande *webring*. Poder-se-ia ter também a divisão de uma grande categoria de rede social, em mais categorias que tratem de assuntos mais específicos.

Deste modo, o problema de pesquisa abordado neste trabalho pode ser assim definido: Como detectar de forma automática relacionamentos implícitos entre escritores de *blogs* (blogueiros), definindo-se possíveis colaboradores, bem como explicitando redes sociais?

1.3 Objetivos

Este estudo tem a finalidade de definir uma ou mais técnicas que permitam agrupar *blogs* e autores por similaridade do conteúdo de suas postagens, e descobrir o grau de afinidade entre os grupos constituídos, elucidando assim conexões que estavam implícitas.

Para o contexto deste estudo, os comentários sobre os posts não são levados em consideração, bem com a lista de seguidores não será avaliada. Considera-se que tais elementos permitem definir afinidades de forma trivial, não fazendo parte do escopo deste trabalho.

Os objetivos específicos a serem atingidos no contexto deste estudo são:

- Estudo do estado da arte referente ao contexto proposto;
- Estudar o processo de pré-processamento que melhor se adapta ao contexto deste estudo;
- Estudar qual algoritmo de *clustering* melhor se adapta ao contexto deste estudo;
- Documentar resultados obtidos, com estudos similares;
- Implementar protótipo de ferramenta de extração e análise de similaridade entre os blogs estudados.

1.4 Estrutura do Trabalho

O trabalho está organizado em mais seis capítulos. O capítulo 2 descreve estudos similares, que tornam esta pesquisa relevante no cenário acadêmico onde se encontra. O capítulo 3 exibe alguns conceitos sobre a área de descoberta de conhecimento que são abordados no estudo. O capítulo 4 aborda outras tecnologias que são utilizadas no estudo, mas que não fazem parte do escopo principal. O capítulo 5 exibe o modelo de implementação proposto, com suas respectivas etapas. O capítulo 6 exibe tópicos relevantes da implementação, resultados obtidos e a validação dos resultados. O capítulo 7 é a conclusão do trabalho, que cita os principais resultados, as limitações do trabalho, trabalhos futuros e percepção do estudo executado.

2 Informações no Ciberespaço

As informações contidas na internet têm sido alvo de estudo, por diversos segmentos acadêmicos e profissionais, pois além de compreenderem uma quantidade muito grande de informações nos mais diversos formatos, culturalmente a sociedade vem sendo afetada pelo advento do ciberespaço, onde virtualmente comunidades, relacionamentos e conseqüentemente cultura tem sido geradas.

2.1 Cibercultura

Conceitualmente, segundo Ramal (2000), a cibercultura é um conjunto das técnicas, práticas, comportamentos e valores que se desenvolvem em torno do ciberespaço, assim como acompanham o seu crescimento, este conceito foi introduzido por Pierre Lévy, professor da Universidade de Paris VIII.

A origem da palavra cibercultura é originada do prefixo “ciber” que vem do grego *Kubernets*, que significa “governar”, o que nos envia à palavra cibernética, que por sua vez foi definida pelo matemático americano *Norbert Wierner*, nos anos 40 do século passado, como sendo a ciência que estuda a comunicação entre homens e máquinas (FERREIRA, 2008).

Hoje, a cibercultura, ou seja, a cultura no ciberespaço, ou no mundo virtual, é uma realidade, que possui vantagens e desvantagens, com relação a avanços, o quesito mais evidenciado é a questão da agilidade na comunicação. Com a chegada dos dispositivos móveis houve também, a alta disponibilidade. Com relação a retrocessos gerados pelo universo cibernético, se destacam a violação de questões básicas, como segurança, privacidade, direitos e deveres (FERREIRA, 2006).

Neste contexto de cibercultura, as tecnologias de comunicação digital que mais se destacam são *blogs*, sites de relacionamentos e serviços de mensagens instantâneas. Na próxima seção será exibido um dos fenômenos criados pelo surgimento da cibercultura, as redes sociais via Internet.

2.2 Redes Sociais

Conceitualmente, as redes sociais são entes formados por dois conjuntos distintos de elementos: atores, representados por pessoas, instituições ou grupos e conexões e suas conexões, que por sua vez formam estruturas de redes de relacionamento (FREITAS, 2008).

No contexto da internet, as redes sociais são representadas por softwares de redes sociais como o *Orkut* e *Facebook*, e por redes de *blogs*, que são criadas através de referências a outros *blogs*. As conexões e interações no ambiente virtual, segundo (RECUERO, 2005), acontecem de maneira semelhante ao mundo real, podendo variar de acordo com o número de interações, intensidade, sentido da relação (bidirecional ou unidirecional).

Os *blogs* surgiram com intuito de ser uma forma de expressão individual na internet, sem a necessidade de conhecer a parte técnica envolvida no processo, e os primeiros foram utilizados apenas como forma de registro de *links* para sites menos conhecidos. Atualmente, os *blogs* são ferramentas poderosas amplamente utilizadas, profissionalmente em muitos. Com a disseminação do seu uso, informalmente, redes sociais foram formadas. Estas redes que assumem os blogueiros e comentaristas como atores. As conexões neste cenário são representadas pelos *trackbacks*, que são *links* de referências simultâneos entre *blogs*, contidos em comentários, ou até na própria postagem, bem como o *blogroll*, que é uma lista de *blogs* sugeridos pelo proprietário do *blog* (PRIMO, 2003).

2.3 Considerações Finais

O volume de informações geradas no âmbito do ciberespespaço, através de milhares de interações feitas a todo instante na internet, influenciaram irreversivelmente o mundo, as culturas, os hábitos. Devido a este fato, é natural que, este universo tenha sido alvo de pesquisas multidisciplinares, vendo-o de diversos ângulos e interesses. No próximo capítulo será abordada a mineração de dados, que é uma área da computação que estuda técnicas de descoberta de padrões em grandes volumes de dados, com intuito de extração de conhecimento.

3 Descoberta de conhecimento

Também conhecida como KDD (*Knowledge Discovery in Database* – Descoberta de conhecimento em bases de dados), surgiu a partir da necessidade da racionalização dos grandes volumes de dados armazenados por grandes empresas, que atualmente armazenam centenas de *terabytes* de dados, com o intuito de responder perguntas como: “O que fazer com todos estes dados armazenados?”, “Como utilizar o patrimônio digital em benefício das instituições?”, “Como analisar e utilizar de maneira útil todo o volume de dados disponível?” (GOLDSCHIMDT, 2005). A KDD pode ser subdividida em três grandes grupos, de acordo com o foco de sua aplicação.

- Desenvolvimento Tecnológico: Este grupo abrange todas as técnicas de concepção, desenvolvimento e otimização de algoritmos de descoberta de conhecimento;
- Execução de KDD: Este grupo contempla a aplicação das técnicas e algoritmos de mineração de dados em bases de dados, com intuito de descoberta de conhecimento;
- Aplicação de Resultados: Este grupo se beneficia dos produtos gerados pelas técnicas de KDD, fazendo uso de modelos gerados pela mineração. Um exemplo de uso comum é o auxílio de tomada de decisão, baseado em indicadores e predições gerados pelos algoritmos da KDD.

Para que um processo de KDD seja iniciado, três elementos devem existir: um conjunto de dados, especialistas sobre os dados que são avaliados e principalmente um objetivo claro a ser alcançado.

Nas subseções deste capítulo, são exibidos dois grandes grupos de técnicas de mineração de conhecimento, a mineração de dados (e um caso mais específico, que é conhecido por mineração de dados em textos, também conhecido por mineração em dados

não-estruturados, ou ainda KDT (*Knowlegde Discovery in Text* – Descoberta de conhecimento em textos).

3.1 Mineração de Dados (*Data Mining*)

A mineração de dados é uma atividade multidisciplinar, originando-se em diversas áreas, onde se destacam a Estatística, a Inteligência Artificial e a Aprendizagem de Máquina.

Uma aplicação de mineração de dados pode ser classificada quanto às ações que devem ser realizadas (TAN, 2009), como:

- Validação de hipóteses postuladas: Analista realiza processo de KDD com o intuito de comprovar ou refutar hipótese;
- Descoberta de Conhecimento: Busca efetiva por conhecimento, a partir de dados abstratos.

A aplicação também pode ser classificada quanto ao seu macro-objetivo, como:

- Preditiva: Busca de modelo preditivo que, baseado em dados históricos, construa modelo que indique valores futuros;
- Descritivo: Cria modelo que descreva os dados atuais como conjuntos de dados compreensíveis pelo homem.

O processo de mineração possui diversas técnicas que podem ser divididas quanto à natureza de sua aplicação. A técnica que será utiliza pelo estudo em questão será a clusterização (técnica de agrupamentos). A seguir estão listados alguns outros exemplos de algoritmos, separados por natureza. Cabe salientar que alguns algoritmos encontram-se em mais de uma natureza, ou seja, são aplicados com mais de um objetivo (HARINATH, 2006).

- Classificação: De acordo com o nome, o propósito dos algoritmos situados nesta categoria têm a tarefa de classificar os objetos analisados. São exemplos de algoritmos desta categoria: Árvore de Decisões e *Naive Bayes*;
- Agrupamento (Clusterização): Os algoritmos contidos nesta categoria têm a tarefa de agrupar os objetos por similaridade. Como exemplos temos o clássico algoritmo K-Means e K-Menóid;
- Seqüência: Esta técnica busca predizer dados futuros, baseados em dados históricos. Como exemplo temos os algoritmos *Time Series*;

- Associação: Este grupo busca analisar a ocorrência de similaridades em grupos de dados. O exemplo clássico de uso deste algoritmo é a aplicação de carrinho de compra virtual, onde produtos são ofertados baseados no comportamento do consumidor, o algoritmo mais conhecido e utilizado é o APRIORI.

3.2 Agrupamento (Clusterização)

A tarefa de agrupamento, de acordo com as categorias citadas acima, é uma tarefa de descoberta de conhecimento descritiva, que tem a finalidade de particionar registros em subconjuntos.

O principal objetivo da tarefa de agrupamento é maximizar a similaridade intra-cluster e minimizar a similaridade inter-cluster. Por ser uma tarefa de indução não supervisionada, diferentemente da tarefa de classificação, que possui rótulos pré-determinados, a Clusterização descobre os rótulos durante o seu processamento.

A análise através de agrupamento exige que os dados estejam agrupados em formato de vetores ou matriz multidimensional. Na maioria dos casos, os algoritmos utilizados por esta técnica, requerem que seja informado o número de clusters a serem retornados.

O processo de clusterização supõe a existência de n pontos de dados $x_1, x_2, \ldots, x_n$, tais que cada ponto pertença a um espaço d dimensional R^d. A tarefa de clusterização destes pontos de dados, separando-se em k *clusters*, consiste em encontrar k pontos m_j em R^d, de tal forma que a expressão seja minimizada (VALE, 2005).

$$\frac{\sum_i min_j d^2(x_i, m_j)}{N}$$

Nesta expressão $d^2(x_i,m_j)$ denota uma distância entre x_i e m_j. Os pontos m_j são denominados centróides ou médias dos clusters.

Os algoritmos de clusterização normalmente trabalham com dados em formato matricial, onde as linhas representam os objetos a serem *agrupados* e as colunas, os atributos de cada objeto, de acordo com a ilustração a seguir, onde n é o número de objetos e p é o número de atributos dos objetos.

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}$$

Dois métodos de clusterização são os mais utilizados, o método de particionamento e o método hierárquico.

- Método de Particionamento: Consiste em dividir em *k* clusters, onde o usuário informa o número *k*;
- Método Hierárquico: Consiste na decomposição hierárquica dos dados, gerando um *dendograma* (árvore que iterativamente divide os dados, até que contenha apenas um elemento), e pode ser criado a partir da abordagem aglomerativa, que executada dos nodos até a raiz (*bottom-up*) ou abordagem divisiva utilizando uma abordagem da raiz para as folhas (*top-down*).

Entre os algoritmos mais conhecidos e utilizados estão o *K-means, Fuzzy K-means, K-modes e K-menoid*, que são exemplos de método de particionamento (GOLDSCHMIDT, 2005).

3.2.1 Método Hierárquico

Nos métodos hierárquicos os dados são particionados inúmeras vezes, formando uma estrutura denominada **dendograma**, que é uma estrutura de aninhamento de nodos, como ilustrado na Figura 1.

Os métodos hierárquicos necessitam de uma matriz que contenha as métricas de distância entre os agrupamentos, essa matriz é conhecida como matriz de similaridades entre agrupamentos. Exemplificando, pode-se considerar um estágio do algoritmo onde o número de agrupamentos é três, e estes são representados por G1, G2 e G3, e cuja a matriz de similaridade entres estes seja representada na Tabela 1. Pelos dados exibidos na tabela acima (que é uma matriz simétrica), é possível afirmar que os agrupamentos G1 e G2 são os mais similares, enquanto os agrupamentos G2 e G3 são os menos similares.

Tabela 1 - Matriz de Similaridade

	G1	G2	G3
G1	0	0,1	0,3
G2	0,1	0	0,4
G3	0,3	0,4	0

Os métodos hierárquicos podem ser divididos em 2 tipos: Métodos Hierárquicos Aglomerativos e Divisivos.

3.2.1.1 Método Hierárquico Aglomerativo

Esta é a abordagem mais simples, que consiste em agrupar inicialmente grupos pequenos e com alto grau de similaridade. Esta tarefa é repetida reiteradas vezes até que um único grupo contendo todos os agrupamentos seja fundido em apenas um *cluster*. Por fim, têm-se menos agrupamentos, com mais elementos e com pouca similaridade entre si (TAN, 2009). A figura abaixo ilustra um dendograma gerado pelo método aglomerativo

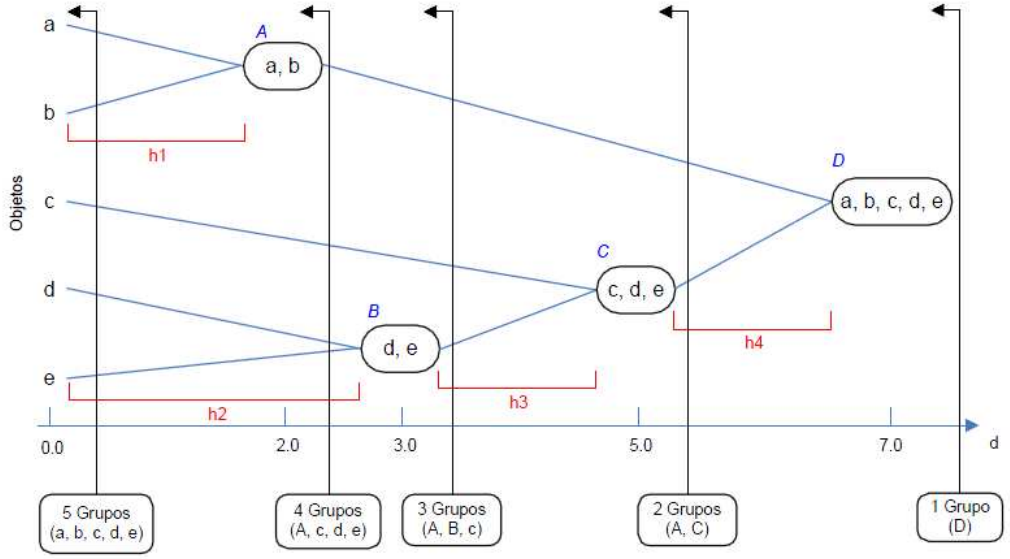

Figura 1 – Método Hierárquico Aglomerativo

Fonte: (Vale, 2005)

Na Figura 1 é possível verificar a execução do método hierárquico aglomerativo, que utiliza uma abordagem *bottom-up*, das folhas até a raiz. No gráfico também é possível avaliar em os pontos A, B, C e D, que ilustram respectivamente o momento das fusão dos agrupamentos (a,b), (d,e), (c,d,e) e por fim a totalidade dos grupos.

Neste contexto, temos mais duas variáveis importantes, que são:

- Coeficiente Aglomerativo
- Banner de Dissimilaridade

O Coeficiente Aglomerativo (CA) é responsável por medir a qualidade de um agrupamento aglomerativo. Sugere que para cada objeto *i*, *d(i)* é sua similaridade em relação

ao primeiro agrupamento em que foi inserido. Dividido pela dissimilaridade final calculada ao final do algoritmo. O Coeficiente Aglomerativo é definido através da fórmula:

$$CA = \frac{1}{n}\sum_{i}^{n} 1 - d(i)$$

Os valores retornados pela fórmula variam entre 0 e 1, e interpreta-se que valores próximos a 0 indicam que nenhuma estrutura foi encontrada e as próximas a 1 indicam que agrupamentos muito claros foram encontrados.

O *Banner* de Dissimilaridade representa as sucessivas uniões entre os agrupamentos (Figura 3). A leitura do *Banner* deve ser feita da esquerda para a direita. Os agrupamentos são dispostos verticalmente e o momento do agrupamento é representado por uma linha horizontal no gráfico. O dendograma e o *Banner* de Dissimilaridade são exibidos respectivamente na Figura 2 e na Figura 3 ilustram os momentos da fusão de dois agrupamentos. Nestas figuras é possível avaliar o momento da união entre os grupos (a, b), (d,e), (c,d,e) e por fim (a,b,c,d,e) (VALE, 2005).

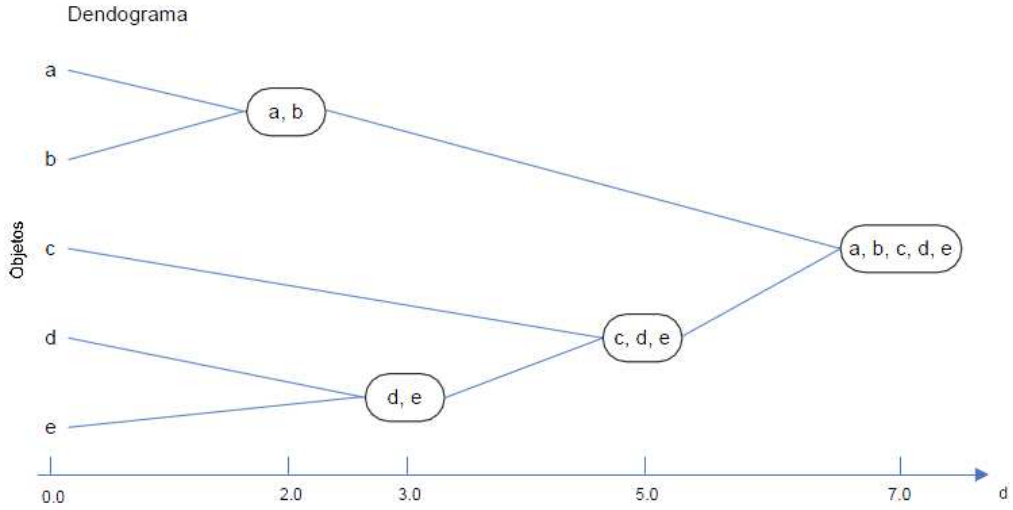

Figura 2 - Dendograma Aglomerativo

Fonte: (Vale, 2005)

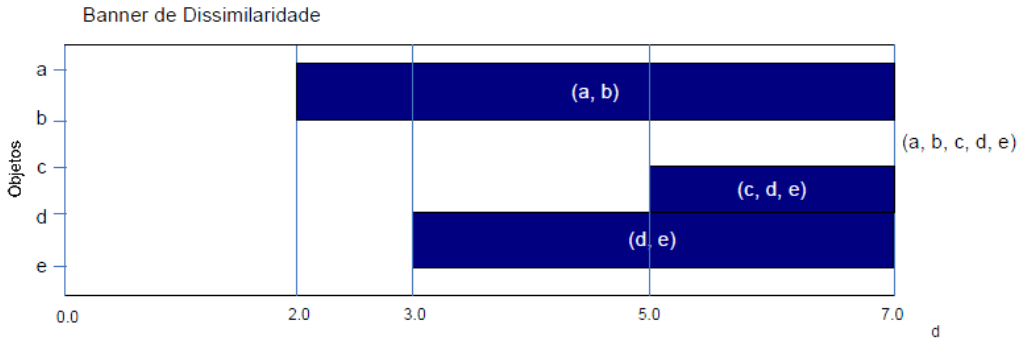

Figura 3 - *Banner* de Dissimilaridade

Fonte: (Vale, 2005)

O *Agglomerative Nesting* (AGNES) é um exemplo de método hierárquico aglomerativo. Suas principais características são a simplicidade das expressões que utiliza e o tempo de computação necessário, que costuma ser muito pequeno.

3.2.1.2 Método Hierárquico Divisivo

Os métodos divisivos são menos comuns, devido as suasineficiências, pois necessitam de um poder computacional maior que os aglomerativos. O método consiste em iniciar com um único agrupamento e a partir deste dividir em vários agrupamentos menores, até que chegue ao ponto que cada agrupamento menor represente apenas um padrão.

O primeiro passo do algoritmo considera todas as divisões de dados possíveis, em dois agrupamentos, prática que deixaria o processamento de um grande número de elementos inviável.

Por outro lado, este método leva vantagem perante o aglomerativo, no que diz respeito à exatidão, pois em seu primeiro passo várias divisões são consideradas, minimizando probabilidade de um erro de decisão (VALE, 2005). A Figura 4 ilustra um dendograma de execução do Método Hierárquico Divisivo.

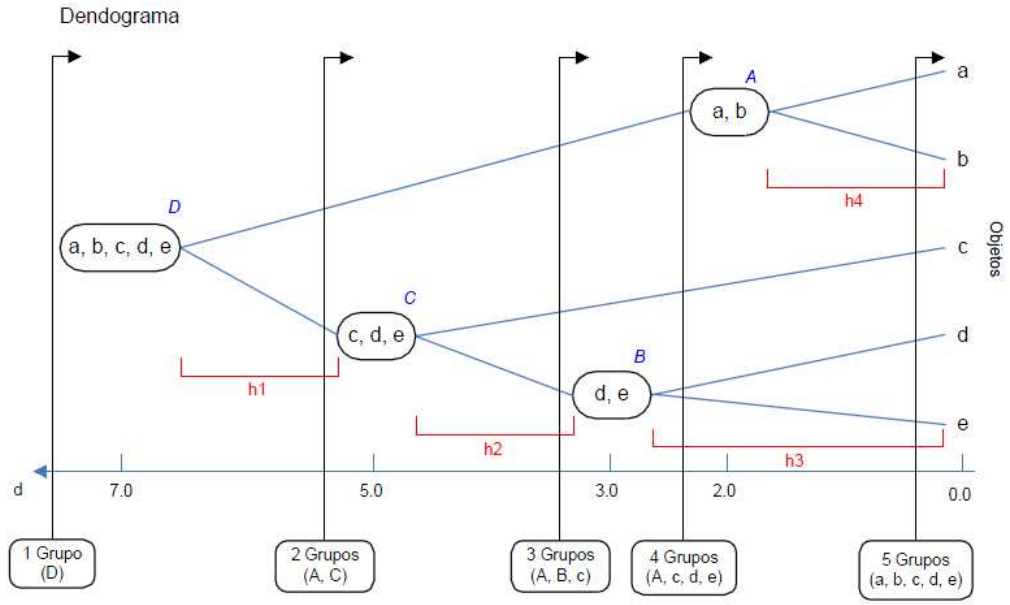

Figura 4 - Dendograma Método Hierárquico Divisivo

Fonte: (VALE, 2005)

Na Figura 4 temos o dendograma divisivo, que comparando com o dendograma aglomerativo da seção anterior (Figura 2), fica evidente a execução pelo caminho oposto. Na ilustração acima os pontos D,C,B e A representam são respectivamente os momentos em que os agrupamentos (a,b,c,d,e), (c,d,e), (d,e) e (a,b) são efetuados.

Neste contexto, temos mais duas variáveis importantes, que são:

- Coeficiente Divisivo
- *Banner* de Dissimilaridade

O Coeficiente Divisivo (CD) é responsável por medir a qualidade de um agrupamento aglomerativo, e sugere que para cada objeto i, d(i) é o diâmetro do último agrupamento ao qual o objeto pertenceu, o Coeficiente Aglomerativo é definido através da fórmula:

$$CD = \frac{1}{n}\sum_{i}^{n} d(i)$$

Os valores retornados pela fórmula acima variam entre 0 e 1, e interpreta-se que valores próximos a 0 indicam que nenhuma estrutura foi encontrada e as próximas a 1 indicam que agrupamentos muito claros foram encontrados.

O *Banner* de Dissimilaridade representa as sucessivas divisões entre os agrupamentos, A leitura do *Banner* deve ser feita da direita para esquerda, os agrupamentos são dispostos verticalmente e o momento da divisão é representada por uma linha horizontal no gráfico. O dendograma e o *Banner* de Dissimilaridade são exibidos respectivamente na Figura 5 e na

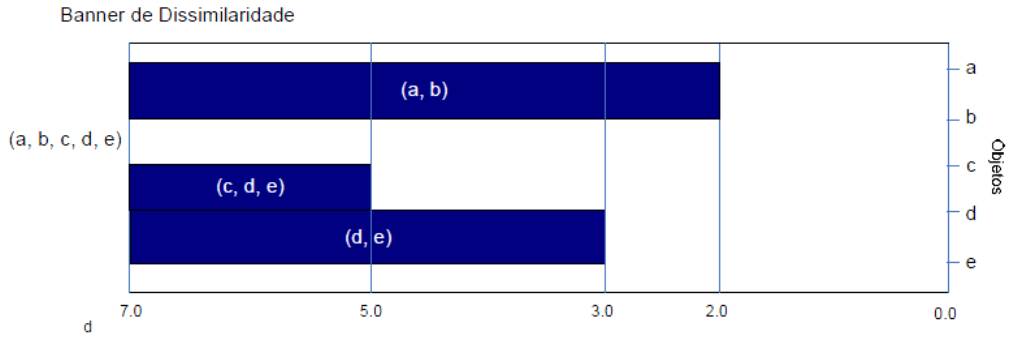

Figura 6, ilustrando os momentos da divisão de dois agrupamentos. Nestas figuras é possível avaliar o momento da divisão entre os grupos (a,b,c,d,e), (c,d,e), (d,e) e por fim (a, b) (VALE, 2005).

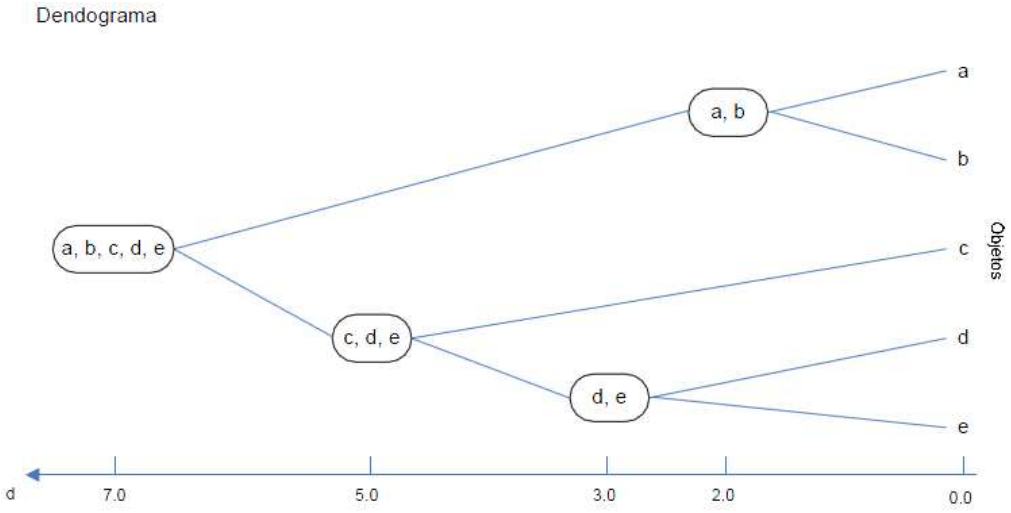

Figura 5 - Dendograma de Método Hierárquico Diviso

Fonte: (VALE, 2005)

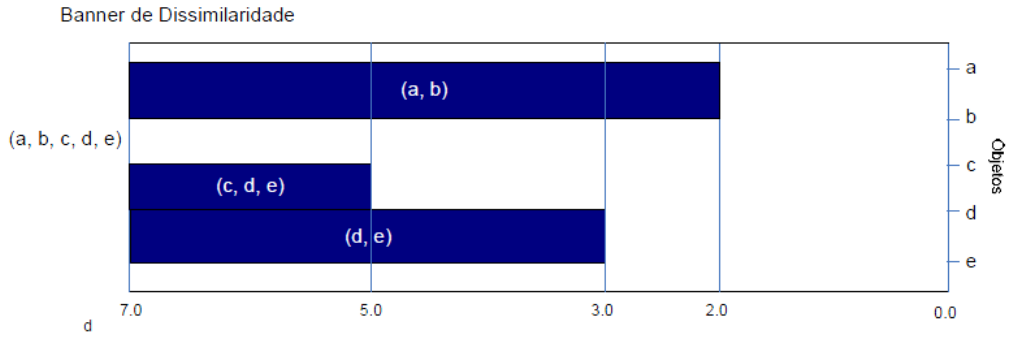

Figura 6 - *Banner* de Dissimilaridade

Fonte: (Vale, 2005)

Neste contexto, temos como exemplos de métodos hoerárquicos divisivo, os algoritmos *Divisive Analysis* (DIANA) e o *Monothetic Analysis* (MONA) como exemplos de método hierárquico divisivo.

3.2.2 Método Particional

O método particional é simplesmente uma divisão de dados em conjuntos não interseccionados, de modo que um objeto esteja em apenas um agrupamento de dados, e também é vedado que um grupo possua subconjuntos, o que então caracterizaria um método hierárquico (TAN, 2009).

Em comparação com os métodos hierárquicos, os métodos particionais possuem a vantagem de trabalhar com bases de dados muito maiores, devido ao seu baixo custo computacional. A desvantagem é que o número de agrupamentos deve ser informando antes do processamento do algoritmo, o que pode implicar em interpretações erradas (VALE, 2005).

Os algoritmos de agrupamento (clusterização) particionais podem ser classificados utilizando as seguintes taxonomias:

3.2.2.1 Métodos Exclusivos

Os métodos exclusivos são caracterizados por criar agrupamentos sem intersecção, atribuindo cada objeto a um único grupo. Neste cenário os algoritmos K-means e K-menoid são exemplos (TAN, 2009).

3.2.2.2 Métodos Não-Exclusivos

Os métodos não-exclusivos, também conhecidos como *fuzzy*, são utilizados para identificar que um objeto pode estar simultaneamente vinculado a mais de um grupo. Possui a vantagem de representar com maior detalhe os objetos, mas esta mesma vantagem pode ser reconhecida como desvantagem, quando a pretensão é encontrar grupos representativos (VALE, 2005).

3.2.3 K-Means e K-Medoid

Este é o algoritmo de agrupamento mais largamente utilizado, no que diz respeito a técnicas de clusterização. O K-Means consiste em, dado um número *k* de grupos, o algoritmo varre os objetos a serem agrupados, agregando os pontos mais próximos e excluindo os mais distantes, baseado na proximidade com os centróides (*K-Means*) ou medóide *(K-Medoid*) calculados.

Os centróides são pontos geralmente não coincidentes, com pontos reais da análise, mas não há nenhuma restrição que o seja. Por sua vez, os medóides são obrigatoriamente representados por um ponto real do conjunto a ser analisado (FONSECA, 2008).

Ambos são classificados quanto à exclusividade de seus grupos, como sendo de agrupamento exclusivo.

3.2.4 *Expectation Maximization* (EM)

O algoritmo *Expectation Maximization* (EM) ou avaliação de probabilidade máxima (MLE), diferente de outros algoritmos de agrupamento, não necessita da informação de quantos *clusters* são gerados após seu processamento. O EM é baseado em distribuições estatísticas, e dividi-se em duas etapas distintas, o E (*Expectation*) e o M (*Maximization*).

Em suas etapas, o algoritmo *EM* é semelhante ao *K-Means*, que possui em sua etapa de *Expectation* a etapa de atribuição de cada objeto a um grupo, diferenciando-se apenas no fato que a etapa de *Expectation* atribui todos os objetos a todos os grupos, associados com a probabilidade que o objeto tem de pertencer a um grupo.

A etapa *Maximization* é similar à etapa de cálculo dos centróides do *K-Means*, mas neste contexto são utilizados todos os parâmetros da distribuição com os seus devidos pesos,

com intuito de maximizar as probabilidades. Em relação à classificação quanto à exclusividade, o EM é um método não-exclusivo.

3.2.5 Índices de Validação de *Clusters*

Os índices de validação de *clusters* são utilizados para validar quantitativamente os resultados obtidos, por algoritmos de agrupamento. No escopo deste estudo três índices foram escolhidos:

O Índice *Davies-Bouldin* consiste em calcular a coesão interna de um grupo e a separação entre eles, a fórmula abaixo exibe o cálculo deste índice, onde *np* é o número de *clusters*,

$$DB = \frac{1}{np}\sum_{i=1}^{np} R_i$$

Onde:

$$R_i = max_{j=1,\ldots,n, i\neq j} R_{ij}$$

E:

$$R_{ij} = \frac{\left(s(C_i) + s(C_j)\right)}{\partial(C_i, C_j)}$$

Sendo que $s: C \rightarrow R$ mede a dispersão interna dentro dos *clusters* e $\partial: C \times C \rightarrow R$ é a distância entre os clusters encontrados.

O objetivo deste índice é minimizar seu valor, e sendo assim, assume-se que quanto menor o valor do índice, melhor é o resultado (CAVALIN, 2005).

O índice de *Silhouette*, ou silhueta, é bastante semelhante ao anterior, e tem a finalidade de definir a qualidade dos agrupamentos avaliando a distância entre os objetos do grupo e avaliando as distâncias entre estes mesmos objetos ao o grupo mais próximo, a silhueta de um objeto é calculada pela fórmula abaixo:

$$silhouette(x_i) = \frac{b(x_i) - a(x_i)}{max\{a(x_i), b(x_i)\}}$$

Onde $a(x_i)$ é o valor médio de dissimilaridade de um objeto x_i comparado aos demais objetos, $b(x_i)$ é média mínima de dissimilaridade entre um objeto x_i e os demais objetos.

Podemos assumir que se um objeto está bem situado no grupo, sua silhueta será positiva. Por outro lado, se esta silhueta for negativa isto indicará que o objeto está mais próximo a outro grupo (FONTANA, 2009).

O coeficiente de *DUNN* é calculado, para identificar o quanto um objeto pertencente a um respectivo agrupamento, e quanto um objeto tem comportamento binário ou difuso, calculado pela fórmula abaixo.

$$F_k = \sum_{i=1}^{N} \sum_{c=1}^{k} \frac{u_{ic}^2}{N}$$

Onde N representa a totalidade de objetos num conjunto, k o número de agrupamentos e u_{ic} é o grau de pertinência do objeto i em relação ao agrupamento c. Utilizando a fórmula citada acima, os valores do coeficiente de *DUNN* podem variar de $1/k$ até 1.

Existe também a versão normalizada do coeficiente de *DUNN*, onde os valores variam de 0 até 1, expressa pela fórmula abaixo.

$$F'_k = \frac{F_k - \left(\frac{1}{k}\right)}{1 - \left(\frac{1}{k}\right)} = \frac{k.F_k - 1}{k - 1}$$

Onde F_k é o coeficiente de *DUNN* não normalizado e k o número de agrupamentos (VALE, 2005). Um valor mais próximo de 1 corresponde a um comportamento mais difuso e um valor mais próximo de 0 corresponde a um comportamento mais binário.

3.3 Mineração de Textos (*Text Mining*)

A relevância da mineração em textos nos últimos tempos, está no fato de que entre os anos de 2006 e 2010, estima-se que aproximadamente 988 *hexabytes* de dados corresponderão aos dados acrescidos no universo digital, e estima-se que deste montante, 80% estará armazenado em formato não estruturado (CONRADO, 2008).

Também conhecida com KDT (*Knowledge Discovery in Text*) é o processo de busca de padrões em dados textuais (MATSUBARA, 2003). Devido ao seu caráter não estruturado, torna-se uma tarefa não trivial, sendo necessária a execução de etapas de preparação do conteúdo, anteriores ao processamento de extração de conhecimento propriamente dito.

A etapa de pré-processamento de textos em linhas gerais, consiste em transformar o conteúdo de um texto em um formato passível de análise. Segundo (MATSUBARA, 2003) existem diversos tipos de representações de texto, mas a mais largamente utilizada é a de *bag-of-words*, que consiste na representação de um documento como um vetor com n posições, onde n é o número de palavras contidas no documento.

Mais detalhadamente, a etapa de pré-processamento pode ser subdividida nas etapas:

- Correção Ortográfica: Etapa de correção ortográfica de palavras no documento, pode ser opcional;
- Remoção de *Stopwords*[2]: Etapa que remove palavras insignificantes, no que diz respeito ao entendimento do texto, e que se repetem inúmeras vezes. São exemplos de *stopwords*, na língua portuguesa, preposições, artigos, interjeições, verbos largamente utilizados, sufixos entre outros. Esta etapa é de suma importância;
- Processo de *Stemming* (radicalização): Esta etapa tem a finalidade de retirada de sufixos de palavras, com intuito de diminuir a quantidade de palavras diferentes para análise.

O processo de *Stemming* está fortemente vinculado à língua em que está redigido o texto, visto que os sufixos avaliados pelo algoritmo de Porter para a língua inglesa, são diferentes e em menos quantidade que o algoritmo RSLP para a língua portuguesa (MONTEIRO, 2006).

3.3.1 Term-Frequency Inverse-Document-Frequency (Tf-Idf)

O *Tf-Idf* é um método que foi criado inicialmente, para resolver problemas de buscas digitais em 1968, e tem sido a técnica mais utilizada para análise de similaridade entre textos, desde então.

O *Tf-Idf* consiste em criar a representação de documentos em formato de vetores, para que a similaridade entre seus conteúdos sejam medidas, sendo que não apenas a tarefa de vetorização é executada para esta representação, existe também a preparação do texto que basicamente divide-se em nas etapas de retiradas de *stopwords* e o processo de *stemming.*

O processo de retirada de *stopwords* consiste em retirar do texto palavras menos relevantes no contexto, que não sejam diferenciais na análise, por serem palavras comuns a qualquer tipo de texto como: artigos, preposições, advérbios, verbos e substantivos comuns, entre outros. Neste contexto é importante salientar que existem listas de *stopwords* disponíveis para uso, e que são freqüentemente incrementadas, e que são dependentes da língua em que os textos são redigidos.

[2] Palavra sem relevância para mineração de textos

O processo de *stemming* ou de radicalização de palavras consiste em remover sufixos irrelevantes para análise textual, processo este que também está vinculado à língua. Nos dois trechos abaixo estão exibidos um parágrafo, contendo o antes e pós processo de radicalização (TORRES, 2004).

> "Sistemas de recomendação são uma excelente solução para a sobrecarga de informações na Internet"
>
> "sistema recom excel soluc sobrecarga inform Internet"

O *Tf/Idf* é calculado em relação ao montante de documentos contidos em uma coleção, logo a relevância de seus termos será calculada em relação a este montante. O *Idf* (*Inverse Document Frequency*) é calculado pela fórmula abaixo.

$$Idf(p) = log\left(\frac{D}{N_p}\right)$$

Onde D é o número de documentos da coleção, e N_p é o número de documentos em que a palavra *p* ocorreu.

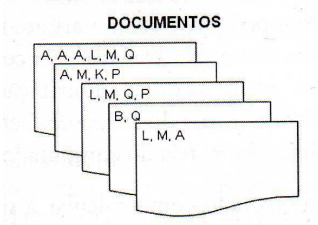

Figura 7 - Coleção de Documentos

Fonte: (TORRES, 2004)

Como exemplo, assume-se que uma coleção de documentos é representada pela figura acima, e abaixo temos um esboço de cálculo e gráfico de *Idf* .

$$Idf(M) = log\left(\frac{5}{4}\right) = 0{,}0969 Idf(A) = Idf(L) = Idf(Q) = log\left(\frac{5}{3}\right) = 0{,}2218 Idf(P) = \log\left(\frac{5}{2}\right) = 0{,}3979 I$$

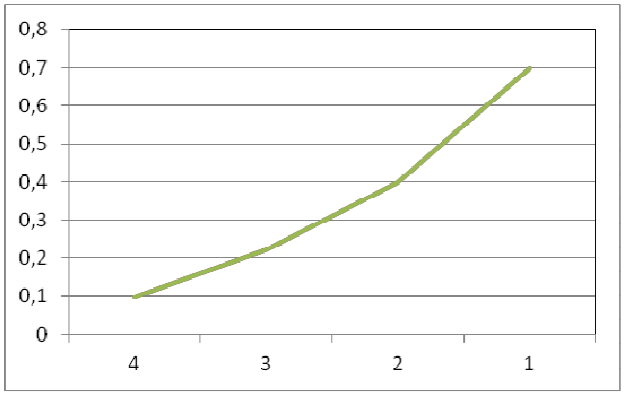

Figura 8 - Gráfico de *Idf*

Fonte: (TORRES, 2005)

De acordo com o gráfico, pode-se perceber que quanto mais rara for a palavra na coleção de documentos, maior será o seu valor de *Idf*.

Após o cálculo do *Idf* para cada palavra, de cada documento, cria-se a representação vetorial de cada documento da coleção. Uma posição *x* do documento corrente representa uma palavra *y*, e no *n*-ésimo documento da coleção a mesma posição *x* conterá dados da mesma palavra *y*, que será preenchida por zero, caso esta não possua a palavra *y* em seu conteúdo. Abaixo está exemplificado o cálculo da vetorização do documento D_1, que representa o primeiro documento ilustrado na figura 7.

$$W(D_1, A) = Idf(A) * Tf(D_1, A) = 0{,}2218 * 3 = 0{,}6654 \quad W(D_1, L) = Idf(L) * Tf(D_1, L) = 0{,}2218 * 1 = 0{,}2$$

Sendo assim, abaixo está a representação vetorial do documento *D1*

$$\overrightarrow{D_1} = (0{,}6654; 0{,}2218; 0{,}0969; 0{,}2218; 0; 0; 0)$$

3.3.2 Similaridade entre Textos

Após a vetorização dos documentos, vista no item anterior, é possível efetuar a análise de similaridade. Neste sentido será utilizado o método do cosseno, pois o produto de sua análise serve de entrada para os passos posteriores, sem nenhuma adaptação posterior.

O método do cosseno consiste em aferir o ângulo entre os vetores, no espaço *n* dimensional, onde *n* é o número de palavras distintas na coleção. A aferição é feita de dois em

dois vetores, e para fins de similaridade, assume-se que quanto mais próximo de 1, mais similares são os documentos, e quanto mais de próximo de 0, menos similares são os documentos.

Para fins de exemplo, os vetores D_1 e Q_1, os quais são representados pelos vetores exibidos abaixo, será o foco da análise.

$$\overrightarrow{D_1} = (0{,}6654; 0{,}2218; 0{,}0969; 0{,}2218; 0; 0; 0) \overrightarrow{Q_1} = (0{,}2218; 0{,}0969; 0; 0; 0; 0)$$

De acordo com o cálculo exibido abaixo, é possível verificar que a similaridade entre os textos representados pelos vetores D_1 e Q_1 é de 0,87, sendo assim é correto afirmar que são muito similares.

3.4 Considerações Finais

O conteúdo deste capítulo busca elucidar o conceito de mineração de dados em dados estruturados e não estruturados, bem como busca explicar algumas técnicas utilizadas. Devido ao foco do estudo proposto, no que diz respeito à mineração em dados estruturados, a técnica utilizada é a de *clustering* (agrupamento), onde o algoritmo escolhido foi o EM (*Expectation Maximization*). Isto deve-se ao fato de ser um algoritmo de agrupamento que não necessita a informação de número de agrupamentos, como os casos mais clássicos de algoritmos de agrupamento como o *K-Means* e *K-Menoid* , cabendo-lhe o papel de encontrar o melhor número de agrupamentos, através de suas características estatísticas.

Segundo (FONSECA, 2008), os resultados produzidos pelos algoritmos podem variar de acordo com diversos fatores, tais como a natureza dos dados, formato dos conjuntos e parametrização do ambiente.

No estudo em questão, a análise de dados não-estruturados foi utilizada dentro da etapa de pré-processamento, onde o método *Tf/Idf* foi utilizado para atribuir pesos à palavras dentro do contexto da coleção de documentos que está sendo avaliada, e para avaliar a similaridade entre pares de documentos.

4 Outras Tecnologias Envolvidas no Estudo

Além dos conceitos e tecnologias apresentados nos capítulos anteriores, mais algumas tecnologias estão envolvidas no estudo corrente. Elas assumem um papel periférico no contexto deste trabalho, mas de são grande valia no que diz respeito à produtividade e usabilidade.

4.1 RIA (*Rich Internet Application*)

É um conceito que prega a utilização das tecnologias disponíveis, a fim de tornar a experiência de uso das aplicações web mais intuitivas e eficientes para os usuários, e a separação de código entre cliente e servidor, ficando a cargo do *client* qualquer processamento de interface, e ao servidor as regras de negócio.

Atualmente, as ferramentas RIA são representadas pelas ferramentas da Adobe como *Flash*, *Flex* e *Air* (http://www.adobe.com/products), Microsoft *SilverLight* (http://www.silverlight.net/) que é um *subconjunto do WPF*, JavaFX (http://www.javafx.com/), OpenLazlo (http://www.openlaszlo.org), e Frameworks baseados em Ajax, como o Google Web ToolKit (http://code.google.com/webtoolkit/).

A quebra de paradigma de desenvolvimento web, com a tecnologia RIA, tem sido a grande barreira para uma adesão maciça dos desenvolvedores, e por enquanto a solução que vem sendo utilizada, é a criação de *Frameworks* que procuram abstrair a complexidade deste novo conceito (GOMES, 2008).

A *Microsoft* está representada no contexto RIA, através do *WPF* e do *Silverlight*, que é um subconjunto do primeiro, ambas baseiam no XAML (*Extensible Application Markup Language*).

O XAML é uma linguagem de marcação baseada em XML, criado pela Microsoft, usada para definir os gráficos de uma aplicação. A *Microsoft* procurando atender padrões impostos pelo mercado, preocupou-se ao criar o XAML em torná-lo um padrão aberto, e na no que tange a portabilidade, criou *plugins* para outras plataformas, como o *Moonlight* (*plugin Silverlight* para plataforma *Linux*).

As páginas desenvolvidas em XAML, são subdivididas em duas partes, sendo a primeira a linguagem de marcação propriamente dita, semelhante ao XML, e a segunda é desenvolvida em qualquer linguagem compatível com o *Framework .Net* 3.0 ou superior.

4.2 UML e Padrões de Projeto (*Design Patterns*)

A UML (*Unified Modeling Language*) é uma linguagem visual, uma notação de desenho com semântica, que é utilizada para representar diagramaticamente os programas computacionais, desenvolvidos no paradigma de Orientação a Objetos (OO). Através da UML é possível visualizar a estrutura e o comportamento dos componentes (objetos) de um programa, bem como a forma que objetos contidos neste programa se relacionam.

A UML em sua versão 2.0 é composta de 13 diagramas, que podem ser divididos em diagramas estruturais e comportamentais.

Os diagramas classificados como **Estruturais** servem para que se possa visualizar, especificar e estruturar as características estáticas do sistema.

- Diagrama de Classes
- Diagrama de Objetos
- Diagrama de Estrutura Composta
- Diagrama de Componentes
- Diagrama de Implantação
- Diagrama de Pacotes

Os diagramas **Comportamentais** servem para representar as características dinâmicas do sistema.

- Diagrama de Casos de Uso
- Diagrama de Máquina de Estados
- Diagrama de Atividades
- Diagramas de Interação
- Diagrama de comunicação
- Diagrama de Visão Geral

Para o projeto, do presente trabalho, foi utilizado os diagramas de classes, componentes e sequência, pois foram necessários para elucidar a arquitetura, estrutura de classes e comportamentos da ferramenta.

Os **padrões de projeto** (*design pattern*) são as melhores práticas adotadas em projetos orientados a objeto, que visam uma melhor estruturação dos projetos e reutilização dos componentes.

Os padrões de projeto podem ser classificados em Padrões de Criação, Estruturais e Comportamentais.

Os padrões **Criacionais** têm como característica a abstração da criação de instâncias de objeto, e são utilizados quase sempre em casos que a criação do objeto não depende apenas de sua instanciação. São exemplos de padrões de projeto **Criacionais** os padrões:

- *Factory Method*
- *Abstract Factory*
- *Builder*
- *Prototype*
- *Singleton*

Os padrões **Estruturais** consistem na representação de estruturas maiores, possuindo diversos objetos e classes, Como exemplos de padrões **Estruturais** têm os padrões:

- *Adapter*
- *Bridge*
- *Composite*
- *Facade*
- *Proxy*

Por fim, os padrões **Comportamentais** prevêem a comunicação entre objetos, representando estruturas de execução complexas. Como exemplos existem os padrões abaixo:

- *Chain of Responsability*
- *Observer*
- *Strategy*
- *Mediator*

Neste trabalho foram utilizados os padrões criacionais *Singleton* e *Factory Method*. O padrão *Singleton* foi utilizado para garantir a instanciação de apenas uma conexão com o banco de dados, e o padrão *Factory Method* foi utilizando com intuito de abstrair a criação de objetos, exibindo apenas as interfaces necessárias para execução do processo.

4.3 Considerações Finais

O conceito RIA está representado no estudo pelo componente de interação com usuário, disponibilizado via *browser*, utilizando a tecnologia Microsoft WPF *Browser Application*, através de *plugin* instalado para a plataforma em questão.

A UML foi utilizada para a modelagem de todo o sistema. Para isso, foi utilizada a ferramenta *Sparx Enterprise Architect 7.5*, onde foram criadas classes, diagramas e a geração de código em *Microsoft C#* (*C Sharp*).

Os padrões de projeto utilizados no sistema foram os padrões *Singleton* e *Factory Method*. O padrão *Singleton* foi utilizado na camada de acesso a dados, com intuito de garantir que exista apenas uma única instância de conexão com banco de dados. Por sua vez, o padrão *Factory Method* foi utilizado com a intenção de abstrair a complexidade, centralizar as chamadas aos componentes a que a interface depende, e principalmente facilitar a extensibilidade em projetos futuros.

No capítulo 6 estão ilustrados e explicados a utilização dos componentes RIA, bem como a utilização da UML e dos padrões de design, no contexto deste estudo.

5 Modelo

O estudo em questão pode ser dividido basicamente nos processos de pré-processamento e análise, e a etapa de processamento fica diluída entre esta duas etapas. A arquitetura utilizada pela ferramenta implementada está propositalmente bem segmentada em sete componentes, explicados em detalhes no capítulo 6, com o intuito de uma fácil extensibilidade para trabalhos futuros.

Nas seções contidas neste capítulo, são exibidos e detalhados os processos e a arquitetura de componentes implementadas por este estudo.

5.1 Pré-processamento

Segundo Matsubara (2003) esta é uma etapa crítica e não trivial do processo, pois é onde ocorrem grandes transformações nos dados, devido ao seu caráter inicial, não estruturado.

Para o escopo do trabalho proposto, cada *post* da página é considerado um texto diferente, isto é, se o *blog* possuir n *posts* em uma mesma página, para fins de extração de dados, são considerados n textos extraídos de um mesmo autor. As *tags* (categorias) também são avaliadas durante o processo, sendo assumidas como uma forma prévia de agrupamento.

O primeiro processo desta etapa é o processo de extração de dados, que tem a atribuição de efetuar a busca de blogs com as características solicitadas. Os conteúdos desta pesquisa são persistidos em repositório de dados, o processo repete a mesma tarefa até que o número de origens solicitadas seja satisfeito.

O segundo processo executa a remoção de *stopwords* dos textos extraídos, esta tarefa consiste em retirar dos textos palavras de pouca relevância no contexto de análise textual e que se repetem inúmeras vezes (Monteiro, 2006). A lista de stopwords disponibilizada em (*SnowBall*, 2009) é utilizada como ponto de partida, para este processo.

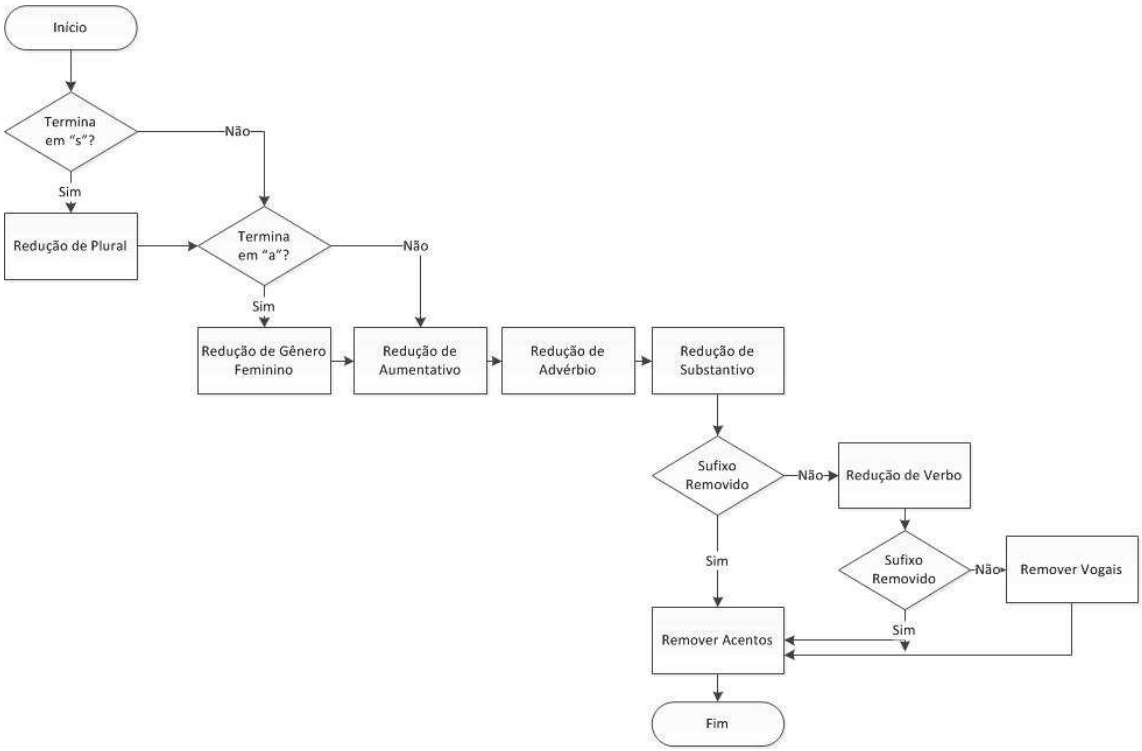

Figura 9 - Esquema do algoritmo RSLP

Fonte: Traduzido de (ORENGO, 2006)

O terceiro passo utilizará a abordagem de *bag-of-words*, que segundo Matsubara (2003) é a mais utilizada, e consiste em transformar os documentos em uma representação vetorial de palavras, para processamento posterior.

Após o processo de vetorização do documento, o processo de radicalização (*stemming*) é iniciado. O processo de radicalização de palavras tem a meta de diminuir a dispersão de palavras no vetor, e para isso é utilizado o algoritmo "Removedor de Sufixos da Língua Portuguesa (RSLP)" descrito em Orengo (2001), ilustrado na Figura 9. Abaixo temos uma breve descrição de cada passo desempenhado pelo algoritmo (Caixas do Fluxograma):

- Redução de Plural: Com raras exceções, na língua portuguesa, a letra "s" é utilizada no final das palavras para representar o seu plural, mas nem todas as palavras terminadas em "s" indicam que estão no plural (ex: lápis).
- Redução do Gênero Feminino: Na língua portuguesa todos os substantivos e adjetivos possuem um gênero, e este passo tem a finalidade em transformar

todos os substantivos e adjetivos do gênero feminino em seu correspondente masculino.

- Redução de Advérbios: Este passo apenas remove o sufixo “-mente” das palavras, que é o indicativo de advérbio (ex: dificilmente).
- Redução de Aumentativos e Diminutivos: Neste passo os sufixos de aumentativo e diminutivo são extraídos do sufixo. Diferentemente da língua inglesa, que possui uma forma mais simples de lidar com estes casos (ex: *small house* = casinha), na língua portuguesa existem as formas diminutivas, aumentativas e superlativas. Enfim, este é um caso que possui diversas exceções.
- Redução de Substantivos: Neste passo o algoritmo substantivos e adjetivos, e se em algum caso o sufixo for retirado, nas etapas posteriores a esta a palavra em questão será desprezada.
- Redução de Sufixos de Verbos: Este passo faz a redução de verbos ao seu radical, enquanto a língua inglesa possui quatro variações, os verbos regulares da língua portuguesa apresentam 50 formas diferentes,.
- Redução de Vogais: Palavras que não foram processadas pelos passos cinco e seis do algoritmo, e possuírem final “a”, “e” ou “o” terão esta vogal retirada.
- Redução de Acentos: Esta redução é efetuada, pois após os passos anteriores ainda é possível termos radicais diferenciados pela acentuação gráfica.

É importante que os passos sejam seguidos na ordem disposta acima, para que não se tenha prejuízo posterior na análise. (ORENGO, 2006).

Segundo Orengo (2006), uma lista de exceções de regras foi utilizada para o auxílio da análise, e durante o estudo palavras foram adicionadas na lista (*Snowball*, 2009).

5.2 Processamento

Como dito anteriormente, da ótica da arquitetura de componentes, a etapa de processamento está dividida entre os componentes de pré-processamento e análise, mas na implementação, esta etapa é divida em três subprocessos.

[UdW1] Comentário: anotaçõ es

O primeiro subprocesso é o cálculo do *Tf/Idf*, para as todas as palavras distintas encontradas na coleção de documentos. Denominam-se documentos os textos extraídos dos

blogs focos do estudo. A quantidade de documentos da coleção é o número de textos obtidos a partir do processo de extração, e as palavras consideradas para fins de processamento, são aquelas que não estão contidas na base de *stopwords* e que passaram pelo processo de *stemming* RSLP. Esta etapa localiza-se no componente de pré-processamento.

O segundo subprocesso é o processo de cálculo de similaridade entre os textos dos *blogs*, através da regra do co-seno. Este processo está localizado na etapa de processamento, devido ao fato de estar fortemente ligado ao processo de *Tf/Idf* e a vetorização de documentos, mas o fruto de seu processamento servirá também para análise, posteriormente. Este é um dos processos computacionais mais onerosos da ferramenta, pois utiliza a regra do produto cartesiano, cruzando todos os textos com todos, buscando a similaridade.

O terceiro subprocesso consiste na geração do arquivo no formato *ARFF* do WEKA, baseado nas palavras e nos textos extraídos dos *blogs*.

5.3 Análise

Nesta etapa do processo inicia-se a tarefa investigativa, com intuito de encontrar padrões. Esta etapa também conhecida como *clustering*, ou ainda agrupamento de dados.

Foi utilizada a ferramenta WEKA, para análise de agrupamentos e visto que o algoritmo escolhido recebe como parâmetro de entrada, o número de clusters, o experimento foi repetido inúmeras vezes em uma mesma massa de dados, onde apenas o número de cluster foi modificado.

A interação entre as duas ferramentas é feita através de arquivos de texto (formato ARRF - formato suportado pelo WEKA). Este arquivo é gerado pelo componente de análise com os dados oriundos do repositório.

Após a execução do método de agrupamento (*clustering*) efetuado pelo WEKA, o componente de análise recupera os arquivos gerados e os torna persistente no repositório de dados.

5.4 Considerações Finais

Este capítulo exibiu quais são as fases de execução do projeto e suas atribuições, como ocorrem às integrações com ferramentas de terceiros (WEKA). O protótipo de ferramenta que implementa o processo exibido neste capítulo é mostrada no capítulo seguinte.

6 Implementação e Avaliação

Neste capítulo, o estudo propriamente dito é explicado. Ele é dividido na seção de implementação, que tem o foco no protótipo, explicando o que, como e porque foi feito, e a seção de avaliação, que aborda os resultados obtidos com o estudo.

Na seção Implementação *são* exibidos alguns objetos componentes da arquitetura da ferramenta, bem como diagramas e algumas de suas funcionalidades, mas apenas em parte. O apêndice A contém os demais diagramas que documentam a ferramenta. Os apêndices também contêm os diagramas de ER do repositório de dados e outras tabelas com dados auxiliares do estudo.

6.1 Implementação

A ferramenta é composta por seis componentes, sendo cinco *DLLs* (*dynamic linked libray*) e um aplicativo web, rodando *WPF*[3] em um ambiente Microsoft IIS 7.0. Na Figura 10 é ilustrado o diagrama de componentes do sistema, onde é possível avaliar o grau de dependência entre os componentes.

O componente **AFS.UserInterface** serve como ponto de entrada do sistema, é através deste componente que o usuário faz sua interação com o sistema. É representado por diretório virtual no servidor hospedado em um servidor web Microsoft IIS 7.0, e o diretório virtual em questão é repositório de páginas com XAML.

Junto com o componente de interface AFS.UserInterface, os componentes AFS.Analise e AFS.PreProcessamento contêm as regras relacionadas ao estudo proposto. Os componentes auxiliares e suas principais atribuições, para que haja um melhor entendimento da arquitetura, podem ser descritos do seguinte modo:

[3] Windows Presentation Foundation - Plataforma RIA da Microsoft, que disponibiliza interface rica em browsers

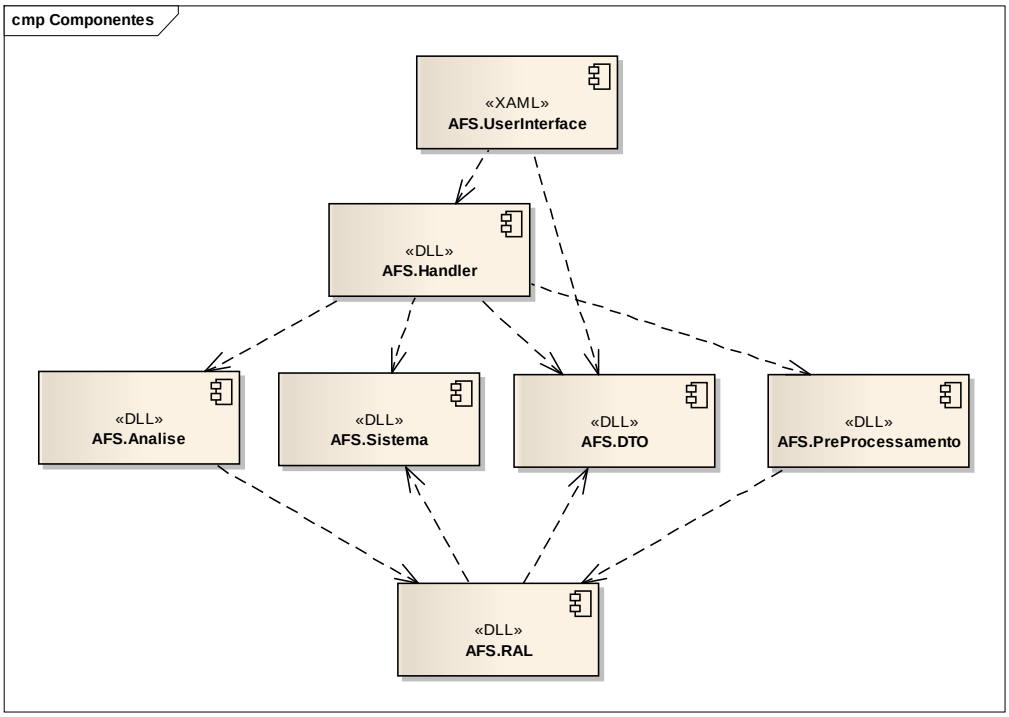

Figura 10 - Diagrama de Componentes

Fonte: Autoria Própria

- **AFS.RAL**: O RAL (*Resource Access Layer*) é responsável por qualquer acesso feito a recursos externos ao sistema. No sistema em questão, o *RAL* faz acessos ao repositório de dados e as páginas na internet, durante o processo de pré-processamento;
- **AFS.Sistema**: Responsável por gravação de *logs* do sistema, preparado para possuir métodos e classes extensíveis.
- **AFS.DTO**: o DTO (*Data Type Object*) é responsável por manter as estruturas de objetos utilizadas em todas as camadas, enumerações, interfaces e eventos;
- **AFS.Handler**: Este componente é a implementação do padrão de *Design Factory Method*, que tem a finalidade de esconder a complexidade do sistema e coordenar a instanciação de objetos (SHALLOWAY, 2004).

A ilustração Figura 11 exibe o fluxo principal da ferramenta, através de um diagrama de seqüência correspondente ao processo completo de extração e análise, exibindo também os

principais subprocessos dos componentes de pré-processamento e de análise, bem como a ordem em que são executados.

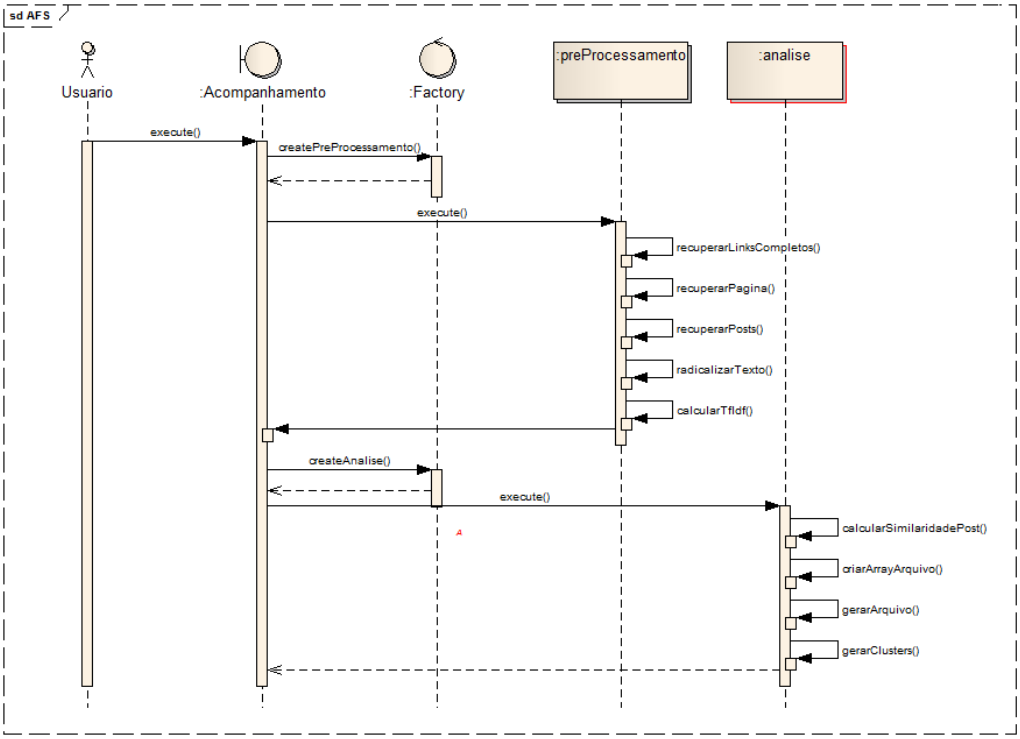

Figura 11 - Diagrama de Sequência

Fonte: Autoria Própria

De acordo com o diagrama da Figura 11, é possível identificar os componentes **Acompanhamento** representado pela notação da UML *boundary* (limite), **Factory** representado pela notação *controller* (controlador), e as classes **preProcessamento** e **Analise**, que responsáveis pelas fases de mesmo nome.

Todos os passos compreendidos pelo diagrama de seqüência da Figura 11 podem ser caracterizados como a seguir.

O **primeiro passo** do diagrama de seqüência, compreendido pela classe/método "Acompanhamento::execute" é executado após o carregamento da página de **acompanhamento.xaml**, que é responsável por disparar o início do processo.

O **segundo passo** representa a classe/método **Factory::createPreProcessamento(),** e através da implementação do padrão *Factory Method*, o método retorna um instância do objeto **preProcessamento**, para que seja possível iniciar a fase de processamento. É

importante salientar a forma de instanciação do objeto, que é feita através da interface atribuída à classe correspondente. Neste estudo tanto a classe **preProcessamento**, quanto a classe **Analise** possuem o mesmo tratamento. O trecho de código abaixo ilustra esta abordagem, e em destaque está exatamente a implementação do padrão *Factory Method*, onde é possível visualizar a classe abstrata ***Factory*** e os métodos estáticos **createPreProcessamento e createAnalise** que o representam.

```
IPreProcessamento p;
IAnalise a;

p = Factory.createPreProcessamento();
p.execute(dias, lista);

a = Factory.createAnalise();
a.execute();
```

Ainda visualizando o trecho de código exibido, podemos avaliar o método **execute** utilizado pelo objeto ***p***, que é responsável por disparar o método de entrada da execução do pré-Processamento e partes da etapa de processamento. Os parâmetros **dias** e **lista,** que são respectivamente os dias retroativos, contados a partir da data corrente, que compreendem o intervalo de tempo a ser pesquisado, e a lista de *urls* que contém os *blogs* a serem pesquisados. Este trecho de código equivale ao **terceiro passo** do diagrama descrito.

Os passos contidos no intervalo de **4** a **8** (recuperarLinksCompletos, recuperarPagina, recuperarPost, radicalizarTexto, CalcularTfIdf) são passos executados internamente no componente de **preProcessamento**, ora em métodos da própria classe, ora em métodos de classes internas do próprio componente.

Para um melhor entendimento da etapa de pré-Processamento, é preciso elucidar o fato que os *blogs* envolvidos no processo são todos hospedados em domínio http://*.wordpress.com (exemplo: http://glaucocortez.wordpress.com), que por sua vez ao informar o endereço do *blog* no formato “http://*.wordpress.com/ano/mes/dia” retorna a lista e um resumo de todas as postagens deste autor no dia informado.

O **passo 4**, representado pelo método **recuperarLinksCompletos** faz requisições *HTTP* via componentes da plataforma *.Net*, buscando todo conteúdo HTML das páginas e persistindo em bases de dados, para posterior análise, fazendo este passo em todos os endereços da lista e no intervalo de dias solicitados.

Quando não há postagens em um determinado *blog*, em um determinado dia, o servidor retorna o erro de requisição *HTTP*, número 404 (*Not Found*), que é guardado em um arquivo de *log*.

Após a extração de todas as páginas solicitadas da internet, o **passo 5** representado pelo método **recuperarPagina**, consiste em avaliar o conteúdo extraído pelo passo anterior, recuperando os *links* para as páginas dos *blogs* com conteúdo na íntegra, e não apenas um resumo. Por fim, munido da lista links de *blogs* íntegros, o método os visita, recupera seus conteúdos e persiste em base de dados.

Terminadas as etapas de extração de dados, é necessário segmentá-los para análise, e para isso o **passo 6,** representado pelo método **recuperarPosts,** faz uma varredura em todo o conteúdo persistido pela etapa anterior, buscando apenas o texto compreendido na postagem. Para isso os *blogs* contidos no *site* do provedor de *blogs WordPress* foi de grande utilidade, pois por mais customizados que fossem os *sites* foco do estudo, algumas marcações HTML do texto sempre serviram para delimitar o conteúdo das postagens, tornando possível extraí-las desprezando comentários, *blogroll*, categorias, entre outras customizações.

O **passo 6** utiliza a biblioteca da *Microsoft.mshtml*, que é uma DOM[4] (*Document Object Model*) de HTML. Sendo assim, é possível encontrar facilmente as estruturas necessárias para extração correta, baseado no comportamento constatado pelas *tags* HTML dos documentos em questão. Após esta extração, apenas o conteúdo da postagem é persistido em base de dados.

O **passo 7,** representado no diagrama pelo método **radicalizarTexto,** é o de radicalização de textos, onde são aplicadas as técnicas de *stemming*. No contexto deste estudo foi escolhido o RSLP**.**

O método, na prática, faz mais do que a radicalização de textos. Sua principal proposta é executar o método interno **retirarCaracteresEspeciais** que, como próprio nome diz, retira do texto todos os caracteres indesejados, deixando no texto apenas letras, e executar o método **retirarSinaisGraficos** que é uma tarefa que está prevista no RSLP. Mesmo assim, alguns sinais gráficos ainda persistiam. Também numa prévia dos *stopword*, foi empiricamente constatado que algumas palavras eram consideradas *stopwords* antes do processo, e após processo de RSLP o mesmo não acontecia. Também empiricamente, foi observada a situação oposta. Por isso após o término da radicalização o método de retirada de *stopwords* é executado novamente, isto ocorria devido ao fato algumas palavras ao ter seu sufixo retirado, transformam-se em palavras de outras categorias.

[4] Padrão W3C que exibe uma estrutura de forma hierárquica, sendo possível navegar por seus objetos, assim como consultar e alterar

É durante o processo de radicalização de palavra, que os documentos assumem as características de *bag-of-words*, onde os documentos entram como textos, e após o processamento, são representadas por vetores de palavras. No contexto deste estudo todas as palavras são guardadas em repositório, para fins de análise, sendo possível avaliar questões como:

- Qual palavra é de qual documento?
- Quantos *stopwords* possuíam um determinado documento?
- Quais regras de radicalização foram aplicadas em quais palavras?

O algoritmo RSLP foi implementado baseado em RIOS (2007), e testado exaustivamente, em diversos ciclos do experimento.

Finalizando os métodos contidos no componente preProcessamento, mas logicamente fazendo parte da etapa de processamento, temos o **passo 8** da seqüência, que é representada pelo método **calcularTfIdf**.

A implementação do método **calcularTfIdf** no componente se resume apenas a chamada do procedimento armazenado **sp_CalculaTfIdf,** a qual devido aos problemas iniciais de *timedout* ocorridos nos primeiros resultados do estudo, é disparada prevendo grande tempo de execução.

Após a conclusão do método **execute** do componente preProcessamento e suas ramificações, o controle da execução retorna ao componente **Acompanhamento**, que de acordo com o trecho de código exibido anteriormente (Figura 11), instancia e executa o método **execute** do componente **Analise,** compreendendo os passos 9 e 10 do diagrama.

Os passos contidos no intervalo de **11** a **14**, são executados internamente pelo componente **Analise**, compreendendo as etapas de processamento e análise.

O cálculo de similaridade é o **passo 11** do processo, e assim como o cálculo de *Tf/Idf*, também é executada através de um procedimento armazenado no banco de dados. Este procedimento consiste em analisar a similaridade postagem a postagem. A partir do produto deste processo, temos o primeiro produto de análise do estudo, que também servirá de base para posterior mineração de dados no WEKA. Sendo assim, este processo reside tanto na etapa de processamento quanto na de análise, e este passo é representado pelo método **calcularsimilaridadePost** do componente **Analise.**

O método representado pelo **passo 12** é o **criarArrayArquivo,** que tem a finalidade de criar em memória um vetor com todas as palavras contidas na coleção de documentos, excluindo *stopwords* e palavras que estejam na *stoplist* (lista de palavras que não são

*stopword*s, mas que foram manualmente excluídas da análise), para auxiliar a geração do arquivo de formato *arrf*, que é utilizado pelo WEKA.

O passo 13 é representado pelo método **gerarArquivo,** que consiste na utilização vetor criado no passo anterior para cruzá-los com dados, da base de dados, gerando o arquivo **blogs.arff**. O arquivo deve ter formato semelhante ao exemplo na Tabela 2, onde o atributo **@relation** é o nome da relação que está sendo listada. Nessa tabela, **@attribute** são os atributos contidos na listagem em questão, podendo assumir diversos tipos, bem como classes, de acordo com o exemplo. Por fim a seção **@data** contém os dados propriamente ditos, que devem vir separados por vírgula, atender aos tipos, de acordo o que foi previsto pelos atributos, e, quando forem classes, devem estar contidas na lista definida anteriormnete.

Tabela 2 - Arquivo ARRF

```
@relation iris

@attribute sepallength numeric
@attribute sepalwidth numeric
@attribute petallength numeric
@attribute petalwidth numeric
@attribute class {iris-setosa,iris-versicolor,iris-virginica}
.....
@data
5.1,3.5,1.4,0.2,iris-setosa
4.9,3.0,1.4,0.2,iris-setosa
4.7,3.2,1.3,0.2,iris-setosa
4.6,3.1,1.5,0.2,iris-setosa
5.0,3.6,1.4,0.2,iris-setosa
5.4,3.9,1.7,0.4,iris-setosa
```

Por fim, o **passo 14**, último do processo, consiste em fazer a integração com o WEKA, submetendo o arquivo *blogs.arrf*, para execução, aguardar o retorno em arquivo texto e ao término da execução, persistir o retorno em base de dados.

A integração com a ferramenta WEKA acontece através de chamada por linha de comando, onde o sistema em questão faz a chamada ao WEKA e aguarda seu retorno que será recebido através de um *buffer* de retorno. O texto na linha de comando executada é semelhante ao exibido abaixo.

```
<Path Java>\java.exe -cp '<Path WEKA>\weka.jar' weka.clusterers.EM -N -1 -I 100 -M 1.0E-6 -S 100
```

Onde *N* indica o número de agrupamentos, *I* o número de iterações a serem feitas para maximizar as probabilidades de agrupamento, *M* é o desvio padrão mínimo e *S* a semente randômica. Para fins de comparação, o mesmo arquivo é submetido ao WEKA, utilizando o algoritmo *K-Means* com o parâmetro de número de *clusters* igual ao número encontrado pelo algoritmo EM.

6.2 Avaliação

A etapa de avaliação foi dividida em duas seções, sendo que a primeira aborda em detalhes os resultados obtidos pelo estudo e a segunda faz uma análise comparativa dos resultados obtidos, com estudos similares.

6.2.1 Resultados Obtidos

Para amostra do experimento, foram selecionados 19 *blogs* hospedados no provedor *WordPress*, sendo que o critério mais relevante escolhido para a seleção dos mesmos, foi escolher *blogs* que abordassem o tema **educação** com alguma intensidade, e que as postagens estivessem compreendidas em um intervalo de datas. Neste sentido os selecionados foram os listados abaixo:

- http://glaucocortez.wordpress.com
- http://jspimenta.wordpress.com
- http://cienteca.wordpress.com
- http://izabelrego.wordpress.com
- http://cyberespacocultural.wordpress.com
- http://gremioadid.wordpress.com
- http://mauricioaraya.wordpress.com
- http://semect.wordpress.com
- http://cristianopalharini.wordpress.com
- http://clinicadotexto.wordpress.com
- http://democracianauspja.wordpress.com

- http://cpantiguidade.wordpress.com
- http://mestradodivulgacaocientifica.wordpress.com
- http://outrajanela.wordpress.com
- http://adolfoboving.wordpress.com
- http://profvaleriamecchi.wordpress.com
- http://gilgiardelli.wordpress.com
- http://santacruzfm.wordpress.com
- http://amaieski.wordpress.com

Nos conteúdos dos *blogs* citados acima, a extração de dados e o processamento na íntegra possuem em números absolutos, os valores listados na Tabela 3, onde é possível verificar que a partir de 19 *blogs*, foram extraídos 173 páginas, que possuem uma postagem por dia, de acordo com a segunda linha da tabela.

Tabela 3 – Resumo do Processamento

Item	Valores
Páginas	173
Postagens	173
Palavras	Total 63556 • 59409 (93%) s*topwords* • 4147 (7%) palavras para Análise o Distribuição quanto à distinção de palavra ▪ 2617 (63%) palavras distintas para análise ▪ 1530 (37%) palavras repetidas análise o Distribuição quanto à aplicação de radicalização ▪ 1191 (29 %) sem aplicação ▪ 2956 (71%) com aplicação

Ainda de acordo com a Tabela 3, 63556 é o número total de palavras encontradas na coleção de documentos, valor que é subtraído em 59409 (93%) palavras inúteis para fins de análise, restando apenas 4147 (75) palavras não classificadas como *stopwords*, para serem analisadas, valores similares ao da literatura, no que diz respeito a extração de dados na

Internet (DRAGUT, 2009). As 4147 palavras restantes também são segmentadas em grupos menores, de acordo com sua distinção perante o grupo. Neste tem-se afirmar que das 4147 apenas 2617 são distintas entre si e 1530 se repetem. Quanto à aplicação de regras de radicalização, tem-se em dois grupos, onde 1191 palavras foram para análise sem nenhuma aplicação de regras de radicalização e 2956 sofreram alterações no processo de radicalização. O mesmo quadro exibe informações percentuais, sendo que alguns percentuais são bastante conclusivos e esperados, de acordo com Dragut (2009), como o fato de 93% das palavras na coleção de documentos serem *stopwords.*

A análise de similaridade utilizando a **regra do cosseno** é uma das fases mais onerosas do processo, mesmo que este processo seja feito por um procedimento armazenado. Para evitar perda de desempenho na comunicação entre banco de dados e aplicação, o processo consiste em fazer um produto cartesiano entre duas instâncias da tabela de postagens, excluindo apenas as relações em que os identificadores forem os mesmos, e os casos em que os pares de analise se repitam em lados opostos do produto cartesiano. No contexto atual o número de iterações necessárias, para obtenção dos cálculos de *tf/Idf* para a totalidade da coleção de documentos é 14878 iterações, este número pode ser encontrado através da fórmula abaixo, onde *p* é número de postagens armazenadas no repositório de dados.

$$\text{Número de Iterações} = \frac{(p * (p - 1))}{2}$$

Os resultados obtidos pela análise de similaridade pela **regra do cosseno** estão dispostos no gráfico na Figura 12, em intervalos percentuais de similaridade, onde o valor disposto no rótulo informativo de percentual indica o intervalo do limite inferior até o valor exibido (ex: 10% representam o intervalo de 5% a 10%). Para fins de legibilidade, o primeiro intervalo, que é o mais populoso, não está representado no gráfico, pois contém 14033 ocorrências.

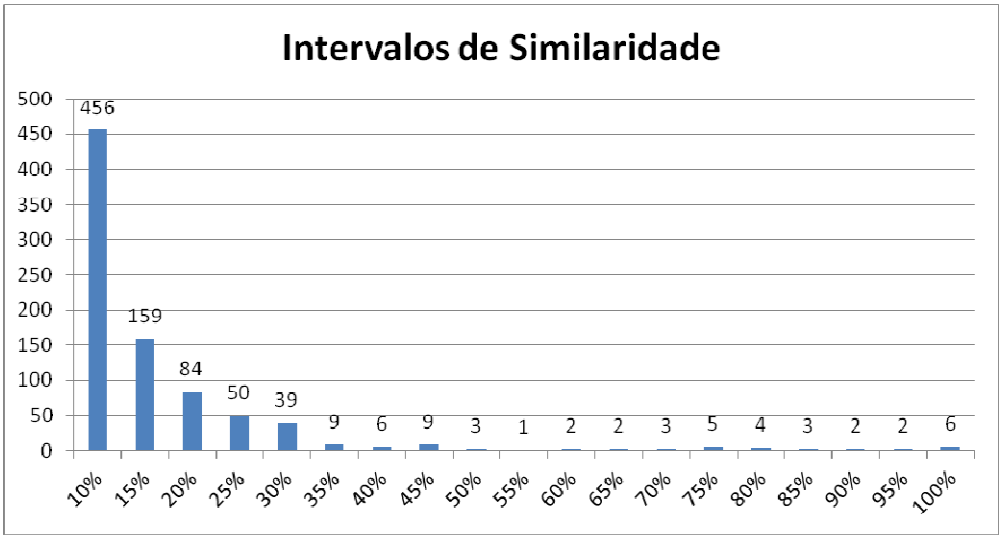

Figura 12 – intervalos de Similaridade

Fonte: Autoria Própria

O gráfico gerado pelos resultados obtidos está de acordo com o esperado, tendo um grande volume de não similaridades. Para um detalhamento maior do estudo, foi feito uma análise mais minuciosa nas similaridades com percentual maior ou igual a 70%, o que indica uma grande similaridade. No gráfico, isto corresponde aos intervalos a partir dos 75%, tendo-se, portanto 22 similaridades para análise.

Inicialmente, foi avaliado quais eram as 6 postagens que eram 100% similares. Neste sentido foi possível avaliar que as postagens envolvidas neste processo de similaridade eram todas do domínio http://semect.wordpress.com/ e possuíam apenas uma imagem em sua postagem. A palavra “semect” foi a única submetida para analise de similaridade, sendo que se considera este um viés de pesquisa. Os *links* que originaram este viés na pesquisa são os citados abaixo.

- http://semect.wordpress.com/2009/10/26/v-olimpiada-brasileira-de-matematica-das-escolas-publicas/
- http://semect.wordpress.com/2009/10/22/seminario-educacao-para-o-mundo-do-trabalho-%e2%80%93-teatro-municipal/
- http://semect.wordpress.com/2009/10/29/i-jornada-de-leitura/

Retirando os seis itens detectados acima, a análise continua sendo feita nos 16 itens restantes. Neste sentido, foi avaliado se o alto percentual de similaridade não está ligado ao fato das postagens pertencerem ao mesmo domínio, mas o resultado foi não conclusivo. De acordo com a tabela abaixo, é possível avaliar o fato de que as 16 ocorrências que estão avaliadas, existem exatamente 8 no mesmo domínio e 8 em domínios diferentes.

Tabela 4 – Comparativo de Domínios

Domínio 1	Domínio 2	Similaridade (%)	Mesmo domínio?
cpantiguidade	democracianauspja	70,39	Não
semect	cristianopalharini	71,02	Não
cpantiguidade	cpantiguidade	71,85	Sim
democracianauspja	cpantiguidade	72,21	Não
cpantiguidade	democracianauspja	72,77	Não
cpantiguidade	democracianauspja	77,16	Não
democracianauspja	cpantiguidade	77,87	Não
cpantiguidade	democracianauspja	78,48	Não
semect	semect	78,59	Sim
gremioadid	gremioadid	82,46	Sim
democracianauspja	democracianauspja	82,86	Sim
cpantiguidade	democracianauspja	83,21	Não
cristianopalharini	cristianopalharini	85,06	Sim
democracianauspja	democracianauspja	88,28	Sim
cristianopalharini	cristianopalharini	90,25	Sim
democracianauspja	democracianauspja	90,62	Sim

Na analise feita através da ferramenta WEKA, utilizando o algoritmo EM (*Expectation Maximization*), através de seu caráter estatístico, chega-se a um montante de sete agrupamentos, onde o de número cinco predomina entre os outros, cabendo-lhe 125 postagens, que na totalidade representa 72%. Outra característica marcante é a diversidade de domínios contidos neste agrupamento.

Pode-se observar também, que exceto o agrupamento número 5, os demais são exclusivos de algum domínio. Ainda é possível verificar que um domínio pode estar em mais de um agrupamento, com é o caso do domínio **cpantiguidade.**

Tabela 5 – Domínios por agrupamento algoritmo EM

Cluster	Domínio	Qtde Blogs
0	democracianauspja	17
1	cristianopalharini	27
2	cpantiguidade	1
3	cpantiguidade	1
4	mauricioaraya	1
5	adolfoboving	1
	amaieski	2
	cienteca	7
	clinicadotexto	1
	cpantiguidade	6
	cristianopalharini	1
	cyberespacocultural	2
	gilgiardelli	1
	glaucocortez	26
	gremioadid	3
	izabelrego	4
	jspimenta	2
	mauricioaraya	11
	mestradodivulgacaocientifica	14
	outrajanela	4
	profvaleriamecchi	2
	santacruzfm	9
	semect	29
6	cpantiguidade	1

Com intuito de comparação, o algoritmo *K-Means* utilizando distância Euclidiana para análise de similaridade foi executado com o mesmo número de *clusters* (*k*) encontrados pelo algoritmo EM. A distribuição de documentos em clusters apresentou algumas divergências, os gráficos exibidos na Figura 13 e Figura 14, ilustram a distribuição dos sete *clusters* encontrados pelos algoritmos EM e *K-Means*. De acordo com a ilustração, as divergências encontradas são pouco sensíveis. Utilizando como exemplo o agrupamento seis, temos uma variação de um ponto percentual e três em valores absolutos.

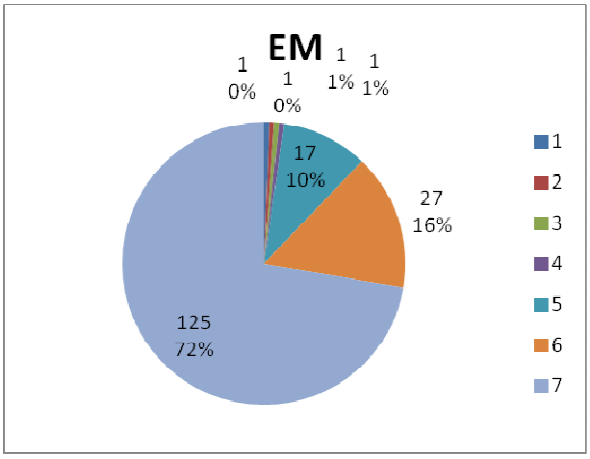

Figura 13 – Agrupamentos EM

Fonte: Autoria Própria

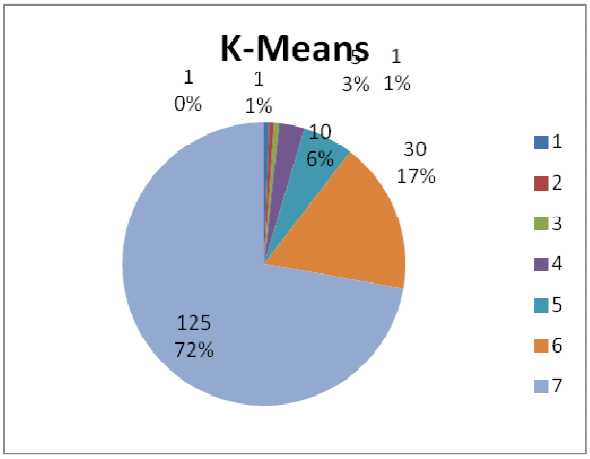

Figura 14 - Agrupamentos *K-Means*

Fonte: Autoria Própria

No que diz respeito à análise e comparação entre algoritmos, mais algumas considerações poderiam ser efetuadas, mas levando em conta o escopo do estudo, as análises efetuadas são suficientes para indicar a viabilidade de aplicar as técnicas propostas no contexto de *blogs*.

6.2.2 Validação dos Resultados

Para avaliar os resultados obtidos, são utilizados os algoritmos de validação de *clusters* (agrupamentos), *Davies-Bouldin*, *Silhouettes* e coeficiente de DUNN, apresentados no capítulo três deste estudo. Os dados foram submetidos à ferramenta *MathWorks Matlab* 7.9.0 R2009b, utilizando o aplicativo *Cluster Validity Analysis Platform (*CVAP*) (Version 3.7)* (KAIJUN, 2009).

O experimento estudado indicou através do algoritmo EM, que a coleção de documentos divide-se em sete agrupamentos. Após a descoberta, os mesmos dados foram submetidos à ferramenta WEKA, utilizando o algoritmo *K-Means* com sete clusters. Os resultados obtidos são bastante semelhantes aos obtido pelo algoritmo EM.

Na ferramenta *Matlab* com o módulo CVAP, entre os dois algoritmos estudados, apenas *K-Means* está disponível. Para fins de análise, foram gerados gráficos avaliando os coeficientes perante o conteúdo e uma variação de número de *clusters*(*k*) de 2 até 9.

A tabela abaixo exibe os dados ilustrados no da Figura 15. Nesta é possível verificar a variação dos valores do índice de *Davies-Boudin*, que mede a coesão intra-grupos e separação intergrupos. O valor do índice para número de agrupamentos encontrados é 2,923 que é quando o valor *k* é igual a 7, sendo que o índice de *Davies-Bouldin* indica que quanto menor o valor, mais o coeso são os grupos encontrados.

Tabela 6 – Índice de Davies-Bouldin

Número de *Clusters* (*k*)	Índice de *Davies-Bouldin*
2	2,0514
3	1,7923
4	3,0808
5	2,3042
6	3,0031
7	2,923
8	1,2508
9	2,0248

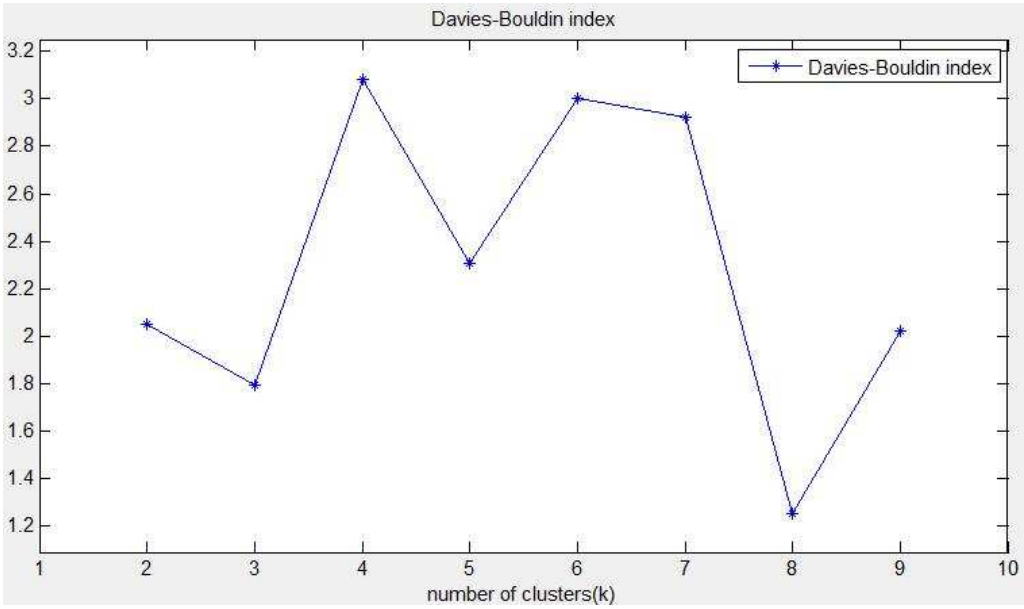

Figura 15 – Gráfico do Coeficiente de *Davies-Bouldin*

Fonte: Autoria Própria

O gráfico da Figura 15 exibe os valores do índice de *Davies-Bouldin*, para o algoritmo K-Means, utilizando a amostra de dados estudada, e variando o número *k* de *clusters* de 2 até 9 *clusters*.

De acordo com os valores tabelados e exibidos pelo índice de *Davies-Bouldin*, quando o número de clusters é igual 8 é que os grupos para o montante de dados em questão ficam mais coesos. Em contrapartida o número de *clusters* encontrados pelo algoritmo de EM apresentam a terceira pior distribuição em relação à amostra de *k clusters* variando de 2 a 9.

Como visto anteriormente no capítulo 3, o coeficiente de *Silhouette* avalia tanto a separação, quanto a coesão de agrupamentos. O coeficiente varia de -1 a 1, sendo que valores próximos a 1 (positivo) indicam uma maior coesão no interior dos agrupamentos e maior separação inter-grupos. Na Tabela 7 estão tabulados os valores referentes aos valores silhueta, para a variação de 2 a 9 agrupamentos, calculados pelo aplicativo *CVAP*, através do Matlab.

Tabela 7 – Valores *Silhouette*

Número de *Clusters* (*k*)	*Silhouette*
2	0.42593
3	0.42787
4	0.44201
5	0.44983
6	0.45732
7	0.44578
8	0.44888
9	0.44213

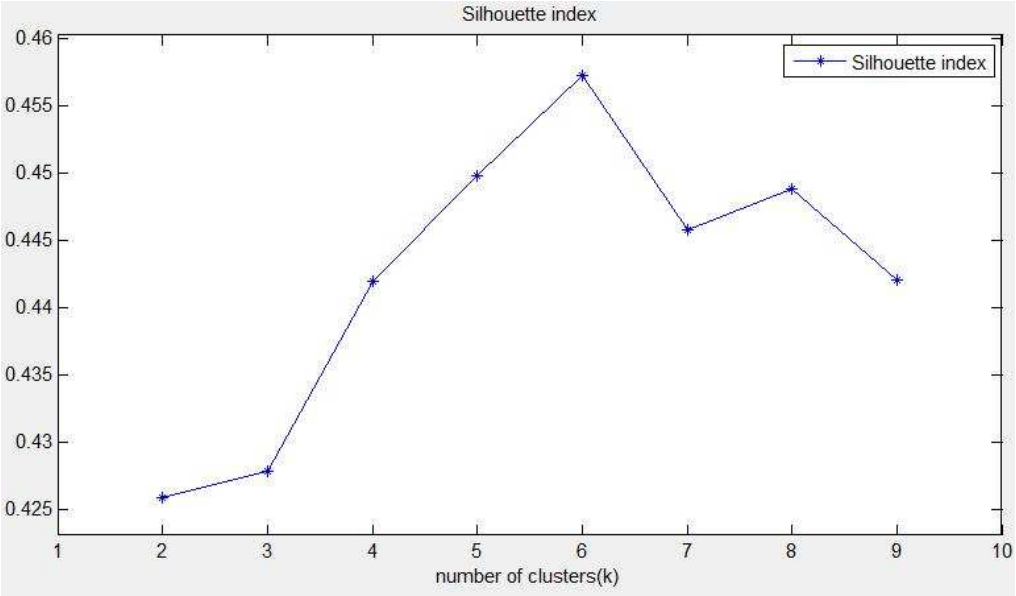

Figura 16 - Gráfico de *Silhouette*

Fonte: Autoria Própria

Assim como o índice de *Davies-Bouldin*, a amostra submetida ao algoritmo de EM não coincidiu com melhor valor de *Silhouette*, ficando o caso encontrado em quarto lugar, mas com uma variação menor que o índice anterior.

Por fim, mediu-se o coeficiente de DUNN que indica se os valores encontrados na coleção de documentos apresentam um comportamento mais difuso ou binário, e neste sentido não apresentou surpresas, pois devido ao grande número de campos contendo zero no vetor-

documento, esta representação era esperada. A tabela 8 e o gráfico da Figura 17 exibem os resultados esperados.

Tabela 8 – Coeficiente de *DUNN*

Número de *Clusters* (*k*)	Coeficiente de *DUNN*
2	0.74078
3	0.74054
4	0
5	0
6	0
7	0
8	0
9	0

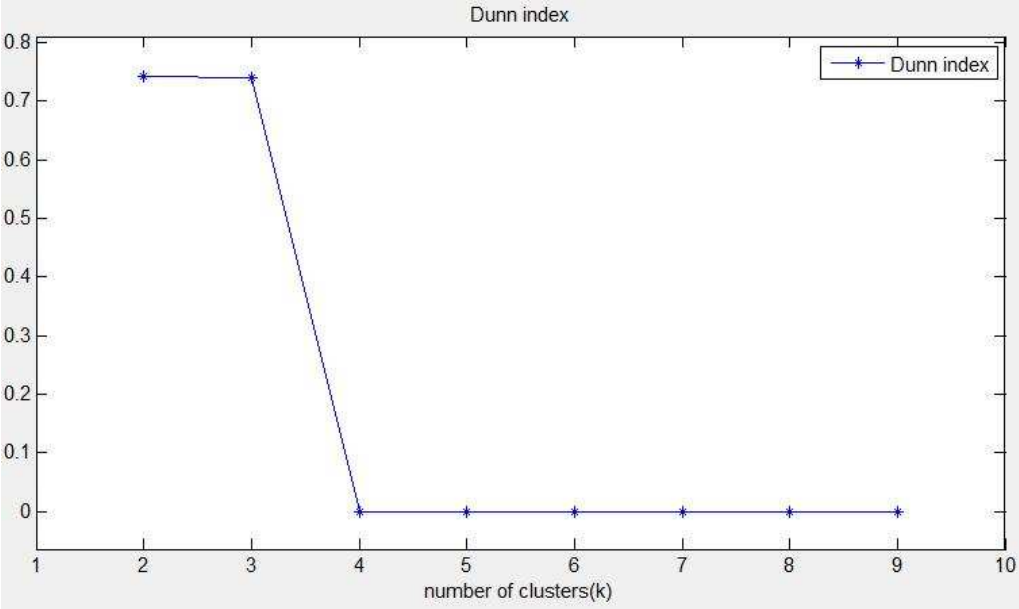

Figura 17 - Gráfico do Coeficiente de *DUNN*

Fonte: Autoria Própria

Para fins de avaliação e análise, foram geradas as distribuições com os números de clusters sugeridos pelo melhor valor do índice de *Davies-Bouldin* e pelo valor de *Silhouette*,

que são respectivamente *k=8* (índice de *Davies-Bouldin = 1,2508*) e *k=6* (*Silhouette = 0,45732*).

Sendo assim, a distribuição feita pelos algoritmos EM e K-Means utilizando seis *clusters* ficou como exibido nos gráficos da **Erro! Fonte de referência não encontrada.**. Neste cenário, assumiu-se o número de clusters sugeridos como melhor caso do índice de *Davies-Bouldin.*

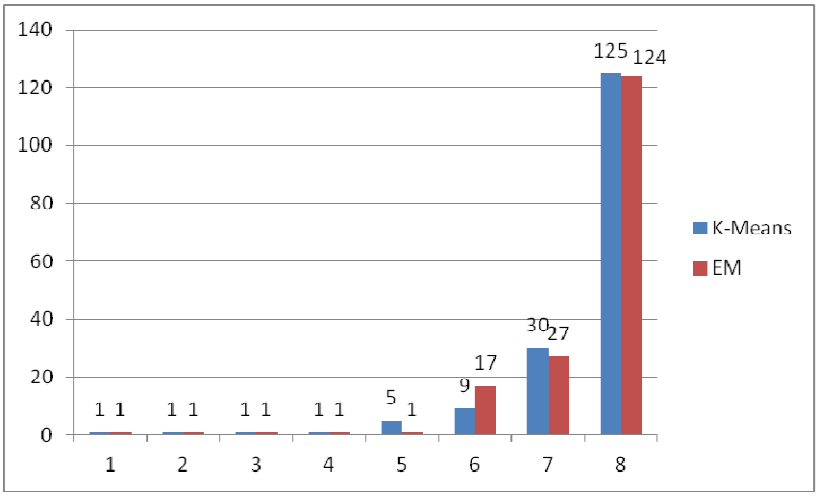

Figura 18 - EM e K-Means em 8 clusters

Fonte: Autoria Própria

Nos gráficos da **Erro! Fonte de referência não encontrada.** e **Erro! Fonte de referência não encontrada.** temos o cenário ideal de acordo com o **valor de silhueta**, que propõe uma distribuição de dados em 6 *clusters*. Os gráficos ilustram esta situação, contemplado os algoritmos EM e K-Means com 6 *clusters.*

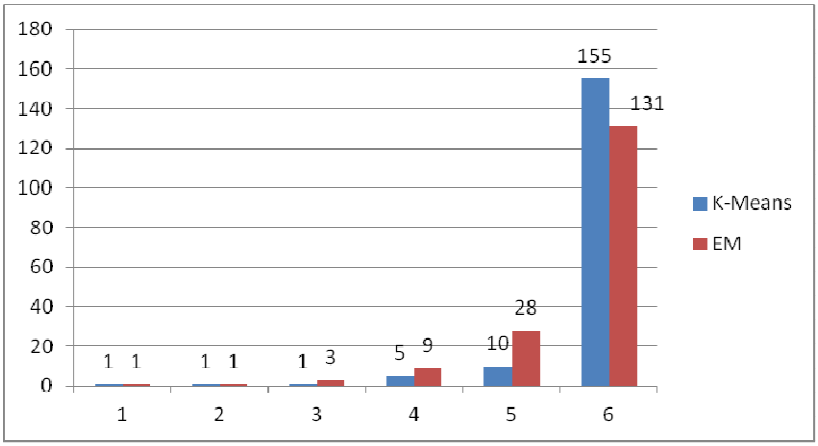

Figura 19 - EM e K-Means com 6 clusters

Fonte: Autoria Própria

Após esta avaliação, e a submissão da coleção de dados para nova mineração com valores de *k* (número de *clusters*) sugeridos como de melhor desempenho pelo índice de *Davies-Bouldin* e Valores de *Silhouette*, a melhor distribuição ficou no experimento com oito *clusters*, Para ter uma melhor visibilidade de amostras maiores de *clusters*, uma extrapolação foi feita utilizando o *MATLAB*, a fim de avaliar o comportamento dos índices citados no estudo. De acordo com o gráfico da Figura 20, é possível avaliar que quanto maior o número de *clusters*, menor tende a ser o valor de silhueta, o que demonstra pouca coesão.

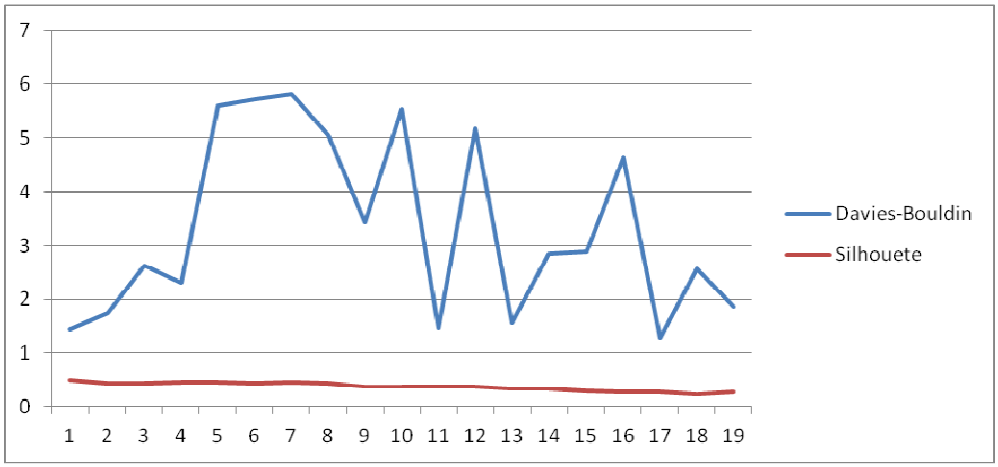

Figura 20 – *Davies-Bouldin* x *Silhouette*

Fonte: Autoria Própria

Comparando os resultados das similaridades encontradas pela regra do cosseno e algoritmo de agrupamento EM, foi possível constatar que das 16 ocorrências encontradas com similaridade igual ou maior 70%, exatamente 8 (50%) ocorrências foram colocadas no mesmo agrupamento pelo algoritmo EM e os 8 (50%) documentos restantes com similaridades altas, foram colocadas em agrupamentos diferentes pelo algoritmo.

Outra constatação é que das ocorrências de alta similaridade encontrada, apenas três são dos documentos residentes no agrupamento que possui 125 ocorrências, e por outro lado este mesmo grupo é onde residem as grandes ocorrências de baixas similaridade.

Por fim, acredita-se que apesar de algumas pequenas discrepâncias de valores entre as técnicas utilizadas, os resultados obtidos pelo estudo apresentam resultados consistentes e similares. Também fica evidente que o processo automático neste contexto ainda não é uma realidade, sendo ainda necessária a intervenção humana.

7 CONCLUSÃO

Ao término do trabalho pode-se constatar o quanto são vastos os assuntos que tratam de processamento de linguagem natural (PLN), redes sociais e ciberespaço e suas respectivas técnicas de busca de conhecimento.

Mesmo com avanços dos algoritmos de mineração, não se consegue uma análise totalmente automática. Um exemplo é o caso citado no capítulo de análise, onde a postagem contida no endereço "http://semect.wordpress.com/2009/10/26/v-olimpiada-brasileira-de-matematica-das-escolas-publicas/" é composta quase que em sua totalidade por uma imagem e mais o nome identificador do site, e isso para fins de análise foi acusado como um viés do processo, apesar de o provedor do serviço de *blogs WordPress* ter sido escolhido através do critério de facilidade de manipulação de conteúdo, programaticamente.

7.1 Resultados Obtidos

Entre os resultados obtidos pelo estudo, pode-se citar a arquitetura que foi planejada para ser extensível. No que diz respeito a provedores de dados. O *WordPress* foi foco do estudo. Ele está contido no componente de pré-processamento, mas a partir de pouco esforço pode ser estendido para outros. Ainda falando de arquitetura de componentes, a interface foi criada com baixo acoplamento em relação ao restante do sistema.

Duas grandes contribuições do estudo realizado foram os processos de extração de conteúdo automatizado, e a integração com o software WEKA, quando comparado a outros estudos.

O grande diferencial da extração de dados não se deve apenas ao desenvolvimento da ferramenta proposta pelo estudo, mas sim pela forma que o provedor de dados *WordPress* disponibiliza as postagens por data, onde recebendo uma url no formato 'http://<end. blog>/aaaa/mm/dd', o mesmo retorna todas as postagens do respectivo *blog* na data informada. Isso facilitou bastante o restante do desenvolvimento e proporcionou a ferramenta a possibilidade de filtrar dados por data, podendo ficar a cargo do usuário a segmentação do intervalo a ser solicitado, e deixando o processo mais automático.

As vantagens em relação ao WEKA residem também no campo da automação, mesmo que em processo diferentes o *framework .Net* possui mecanismos de chamadas a processos externos sincronamente, recuperando seu resultando em *buffer* de memória, pronto para ser manipulado com texto pela ferramenta e persistido no repositório de dados.

Enfim, as maiores vantagens residem no que diz respeito a automatização de processos a recursos externos a ferramenta.

7.2 Limitações

Como limitações do estudo, vejo alguns assuntos que na proposta inicial ficaram fora do escopo, devido à extensão do assunto. Entre os assuntos não abordados no estudo, questões relativas a desempenho, paralelismo e distribuição de processamento são bastante sensíveis, pois em alguns momentos do processo principal, de acordo com o volume que está sendo estudado, o processo pode levar horas em seu processamento.

No que diz respeito à consistência de dados, a limitação do estudo proposto, é que apesar das origens de dados terem sido escolhidas, baseadas no assunto **educação**, a questão ontológica do valor de uma palavra de acordo com o seu contexto não foi levada em consideração.

7.3 Trabalhos Futuros

Este estudo que, em parte é integrante do mestrado profissional multidisciplinar ministrado pelo Centro Universitário La Salle, chamado de “Memória Social e Bens Culturais”, deve ser continuado, com avaliações mais minuciosas e a implementação de quesitos não levados em consideração neste momento, como a técnica de *smoothing*, que consiste atribuição de um valor com tendência a zero, quando o valor de *Tf/Idf* for zero, com o intuito de encontrar valores intermediários no Coeficiente de *DUNN* e avaliar termos perante uma estrutura ontológica, como a do *Thesaurus*,. Também a combinação de duas técnicas de mineração de dados, para obter uma maior confiabilidade no resultado.

Outra proposta de continuação deste estudo é a criação de sistemas de recomendação, a fim de avaliar o modelo, propondo redes sociais.

Neste próximo estudo pode-se avaliar a aplicabilidade do modelo, com avaliações dentro da área de computação e multidisciplinares.

REFERÊNCIAS

RECUERO, Raquel C. (2005) "**Um estudo do capital social gerado a partir de redes sociais no Orkut e nos weblogs**" - Revista da FAMECOS, n. 28, p. 88-106, Dez. 2005.

PRIMO, Alex Fernando Teixeira; RECUERO, Raquel da Cunha. "**Hipertexto Cooperativo: Uma Análise da Escrita Coletiva a partir dos Blogs e da Wikipédia**". Revista da FAMECOS, n. 23, p. 54-63, Dez. 2003.

FONSECA, Raphaela (2008) – "**Uma estratégia de apoio à seleção de algoritmos de clusterização de dados**" – Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia - disponível em http://www.des.ime.eb.br/dissertacoes/

SNOWBALL. Baixado em 08/2009, em http://snowball.tartarus.org/portuguese/voc.txt

MATSUBARA, Edson T.; MARTINS, Claudia A.; MONARD, Maria C. (2003) "**PreText: uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words**" LABIC (Laboratory of Computational Intelligence - USP) disponível em ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/RT_209.pdf

FREITAS, Carla M. D. S.; NEDEL, Luciana P.; GALANTE, Renata; LAMB, Luís C.; SPRITZER, André S; FUJII, Sérgio; OLIVEIRA, José Palazzo M.; ARAÚJO, Ricardo M.; MORO, Mirella M. (2008) "**Extração de Conhecimento e Análise Visual de Redes Sociais**" – Anais do XXVIII Congresso da SBC – SEMISH (Seminário Integrado de Software e Hardware)

MONTEIRO, Lêda de Oliveira, GOMES, Igor Ruiz, OLIVEIRA, Thiago (2006) "**Etapas do processo de Mineração de Textos – uma abordagem aplicada a textos em Português do Brasil**" - Anais do XXVI Congresso da SBC - WCOMPA (Workshop de Computação e Aplicações)

MENDES, Francielle M. M. (2008) "**Blog Pessoal: a busca da identidade do sujeito no mundo mediado pela internet**" - Revista Contrapontos, Vol. 8, No 2 (2008) – disponível em http://siaiweb06.univali.br/seer/index.php/rc/index

CONRADO, M., MOURA, M, MARCACINI, R., REZENDE, S. (2008) "**Avaliando Diferentes Formas de Geração de Termos a partir de Coleções Textual**" – LABIC – (Laboratory of Computational Intelligence – USP – São Carlos) - disponível em http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_334.pdf

SILVA, Jan A. (2003) "**Weblogs: Múltiplas utilizações e um Conceito**" - XXVI Congresso Brasileiro de Ciências da Comunicação – BH/MG – 2 a 6 Set 2003

ORENGO, Viviane M. (2006) "**A study on the use of Stemming for Monolingual Ad-Hoc Portuguese Information Retrieval**" – Cross Language Evaluation Forum (2006) – disponível em http://www.clef-campaign.org/2006/working_notes/

GOLDSCHIMIDT, Ronaldo, PASSOS, Emmanuel (2005) “**Data Mining - Um Guia Prático**” - Elsevier Editora, 2a tiragem

TAN, Pang-Ning, STEINBACH, Michael, KUMAR, Vipin (2009) “I**ntrodução ao Data Mining - Mineração de Dados**” - Editora Ciência Moderna

GOMES, Jaydson M., MACHADO, Rodrigo P. (2008) “**RichBlocks - Um Framework para Implantar Interfaces RIA em Sistemas Web**”

SHALLOWAY, Alan, TROTT, James (2004) “**Explicando Padrões de Projeto - Uma Nova Perspectiva em Projeto Orientado a Objeto**”

VALE, Marcos N. (2005) “**Agrupamentos de Dados: Avaliação de Métodos e Desenvolvimento de Aplicativo para Análise de Grupos**”

TORRES, Roberto “**Personilazação na Internet - Como descobrir os hábitos de consumo de seus clientes, fidelizá-los e aumentar o lucro dee seu negócio**”

RIOS, Davi (2007) "**Stemmer for Portuguese 1.0**" - Implementeção em Ruby do algoritmo RSLP, disponível em http://webscripts.softpedia.com/script/Internet-Browsers-C-C/stemmer-for-portuguese-17904.html

KAIJUN, Wang, BAIJIE, Wang, PENG, Liuqing. "**CVAP: Validation for Cluster Analyses.**" Data Science Journal, Vol. 8, pp.88-93, 2009

CAVALIN, Rodrigo C. (2005) "**Um método para segmentação e reconhecimento de palavras manuscritas usando modelos escondidos de Markov**"

FONTANA, André, NALDI, Murilo C. (2009) "**Estudo e Comparação de Métodos para Estimação de Números de grupos em Problemas de Agrupamentos de Dados**"

RAMAL, Andrea Cecilia. “**Avaliar na cibercultura**”. Porto Alegre: Revista Pátio, Ed. Artmed, fevereiro 2000.

FERREIRA, Aletéia (2008) "**CIBERMODA E SUAS INFLUÊNCIAS NA CIBERCULTURA - A moda do punk ao estilo Matrix**"

FERREIRA, Alexandra (2006) "**Orkut - Reflexo da cibercultura**" - Grupo de trabajo: Cibercultura y nuevas tecnologías de la información. IX Congreso IBERCOM - Sevilla-Cádiz, 2006.

HARINATH, Sivakumar, QUINN, Stephen R., "**Professional SQL Server - Analysis Services 2005 with MDX**", Editora Wrox, Coleção Programmer to Programmer

DRAGUT, Eduard, FANG,Fang, SISTLA, Prasad, YU, Clement, MENG, Weiyi (2009) "**Stop Word and Related Problems in Web Interface Integration**"

Apêndice A – Arquitetura

Este apêndice exibe mais alguns diagramas estruturais da UML, procurando elucidar a forma como a arquitetura foi concebida. A arquitetura da ferramenta, além de estar segmentada em componentes fisicamente, logicamente está disposta em *namespaces*[5]ou espaços de nome.

Os *namespaces* mais importantes criados para organização do estudo estão ilustrados na figuras abaixo, que contém os diagramas de classes dos respectivos *namespaces*, exibindo classes, interfaces, enumerações, métodos e propriedades.

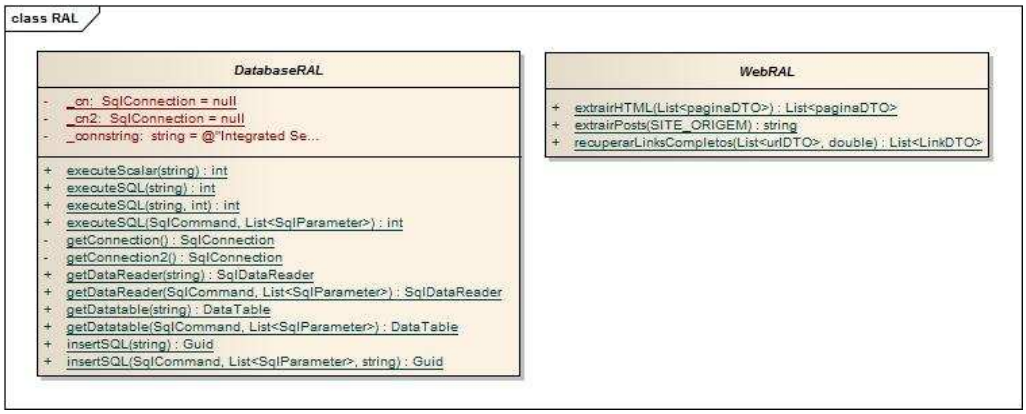

Figura 21 – *Namespace* AFS.RAL

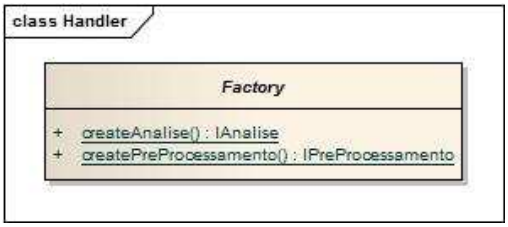

Figura 22 – *Namespace* AFS.Handler

[5] Contexto lógico que organiza objetos de características semelhantes

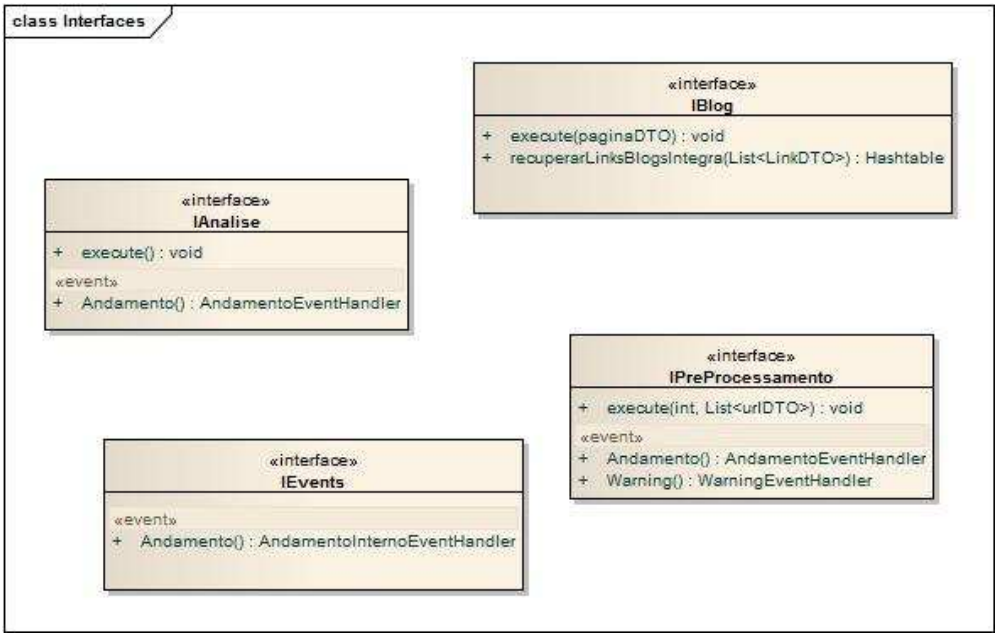

Figura 23 – *Namespace* AFS.DTO.Interfaces

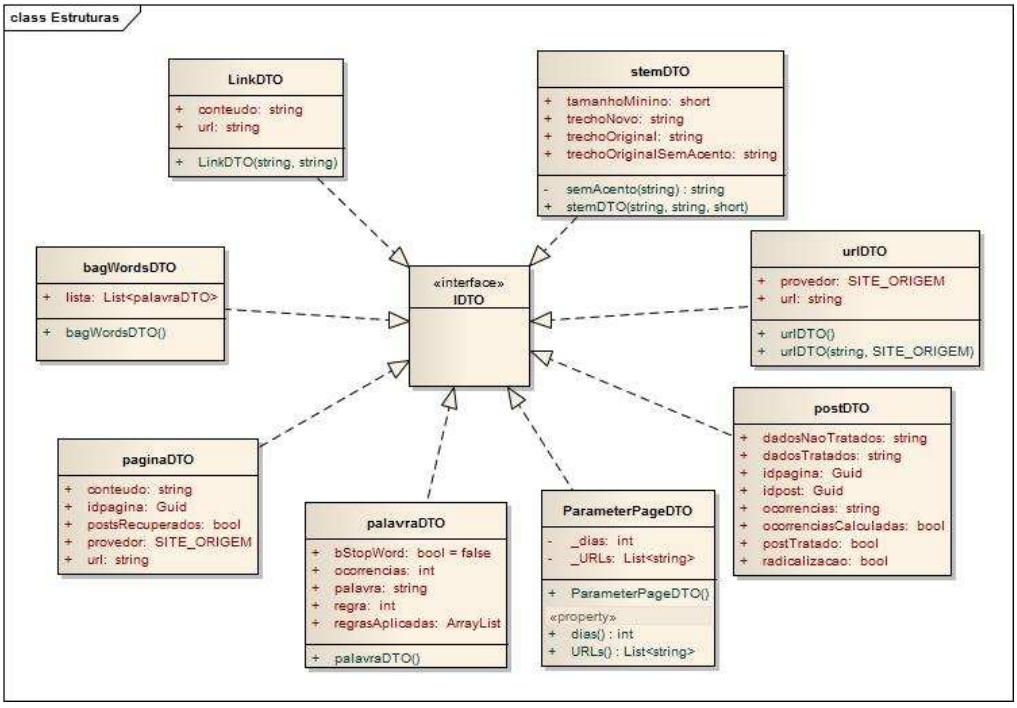

Figura 24 – *Namespace* AFS.DTO.Estruturas

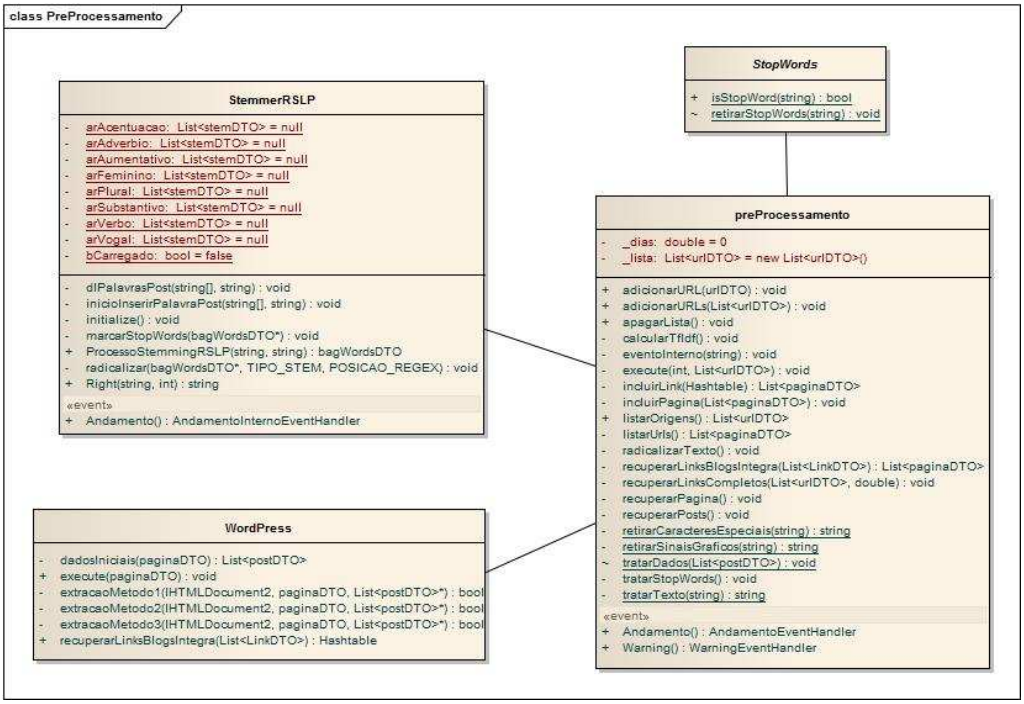

Figura 25 – *Namespace* AFS.PreProcessamento

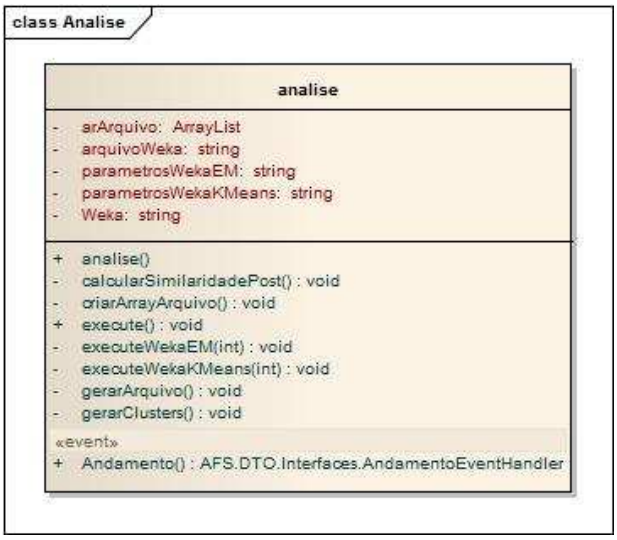

Figura 26 – *Namespace* AFS.Analise

Apêndice B – Modelo ER

O modelo de entidade-relacionamento utilizado como repositório da ferramenta está ilustrado na Figura 27, onde se pode ver a entidades e como se relacionam, sendo as tabelas **pagina**, ***post*** e **palavra** as entidades mais importantes para o estudo, visto que a entidade pagina mantém dados sobre a primeira extração de dados baseados nas páginas de resumo diário dos *blogs*, a entidade ***post*** armazena dados dos conteúdos das postagens extraídas na íntegra, por fim a entidade **palavra** armazena os dados oriundos das postagens, já num esquema vetor-documento, e é a partir desta que os arquivos de integração com WEKA são criados.

As demais tabelas periféricas, são a de ***stopword*** que armazena dados referentes ao conteúdo adquirido em (SNOWBALL, 2009), a de **palavraUnica** que neste contexto de estudo assume o papel das *stoplists*, que são listas que mantém dados que não devem ser considerados para fins de análise de similaridade, mas que não constam da lista de *stopwords*.

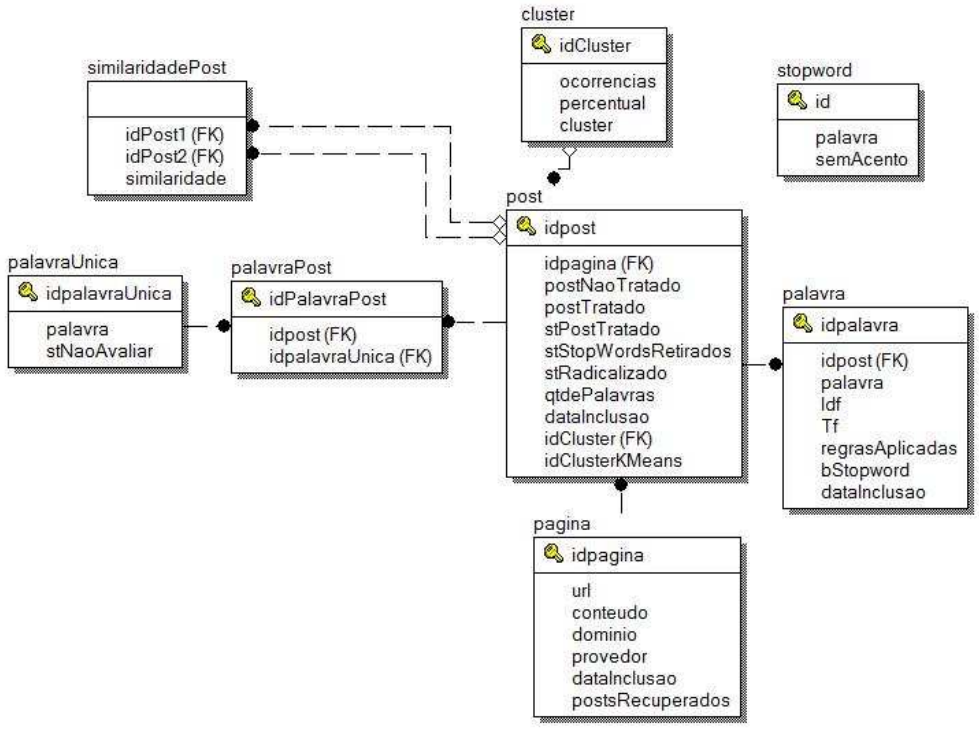

Figura 27 – Modelo ER

As tabelas **similaridadePost** e **Cluster** armazenam respectivamente dados referentes à análise de similaridade utilizando a **regra do cosseno (*Tf/Idf*)** e dados retornados pelo WEKA, contendo cada cluster que foi criado.

Como convenção, form utilizados os prefixo “st” que indica status do processamento e “id” que indicam o identificador único.

Além dos objetos documentados pelo modelo, alguns procedimentos armazenados auxiliares ao processo foram utilizados.

Apêndice C – Telas do Sistema

Este apêndice contém as telas do sistema implementado no estudo, a Figura 28 ilustra a tela inicial do sistema, onde as opções “Novo Processamento” e “Dados Processados” são exibidos. A opção “Novo Processamento” remete o sistema a um novo processamento, onde o fluxo de pré-processamento, processamento e análise são executados, de acordo com os parâmetros ilustrados na Figura 29.

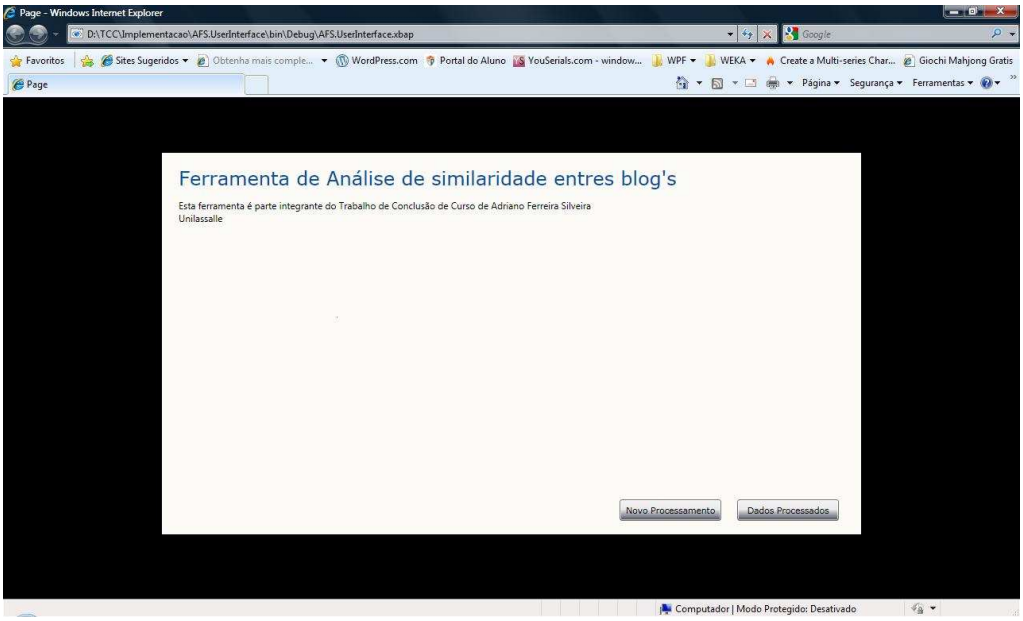

Figura 28 - Tela Inicial

A tela de configuração exibida na Figura 29 exibe os parâmetros de configuração do sistema. O parâmetro número de dias indica o número de dias retroativos ao dia atual, que determina o intervalo de extração de dados, a lista de endereços é preenchida com endereços do provedor *WordPress*, que intencionalmente foi deixado fixo, por ser o provedor foco do estudo. O botão “Padrão” carrega os campos de dia e lista com dados arbitrados como padrão no estudo.

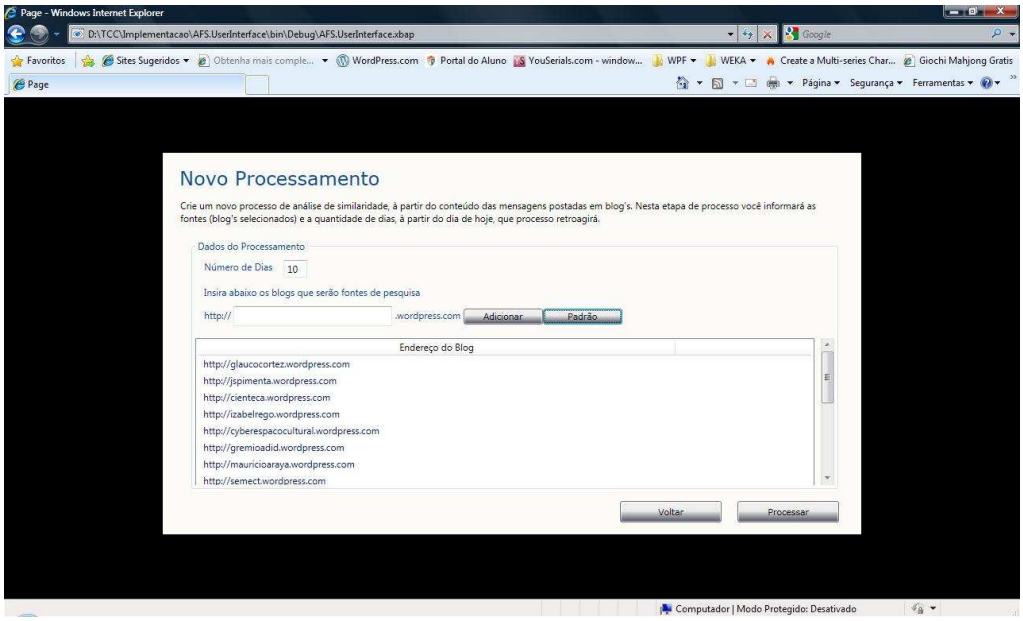

Figura 29 - Tela de Configuração

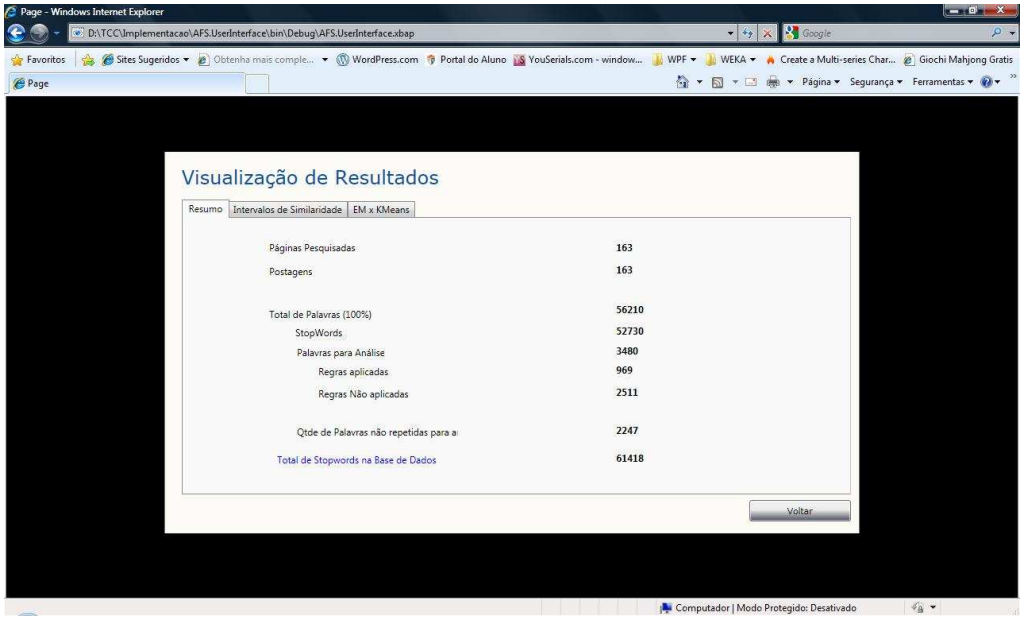

Figura 30 - Visualização de Resultados (Resumo)

A exibição da tela visualização de resultados (Figuras 32, 33 e 34) ocorre ao selecionar a opção “Dados Processados” da tela inicial (Figura 28) ou ao término de um processamento. A aba “Resumo” desta tela exibe o resumo do processamento, como o total de palavras

extraídas para estudo, o número de *stopwords* encontrados, palavras disponíveis para análise, palavras que tiveram e não tiveram regras aplicadas, entre outras.

A aba “Intervalos de Similaridade” exibida na Figura 31 ilustra a quantidade de similaridades entre as postagens, dividas de intervalos de 5 em 5%, sendo que os dois primeiros intervalos não são exibidos, para haja facilidade na leitura.

Figura 31 - Visualização de Resultados (Intervalos de Similaridade)

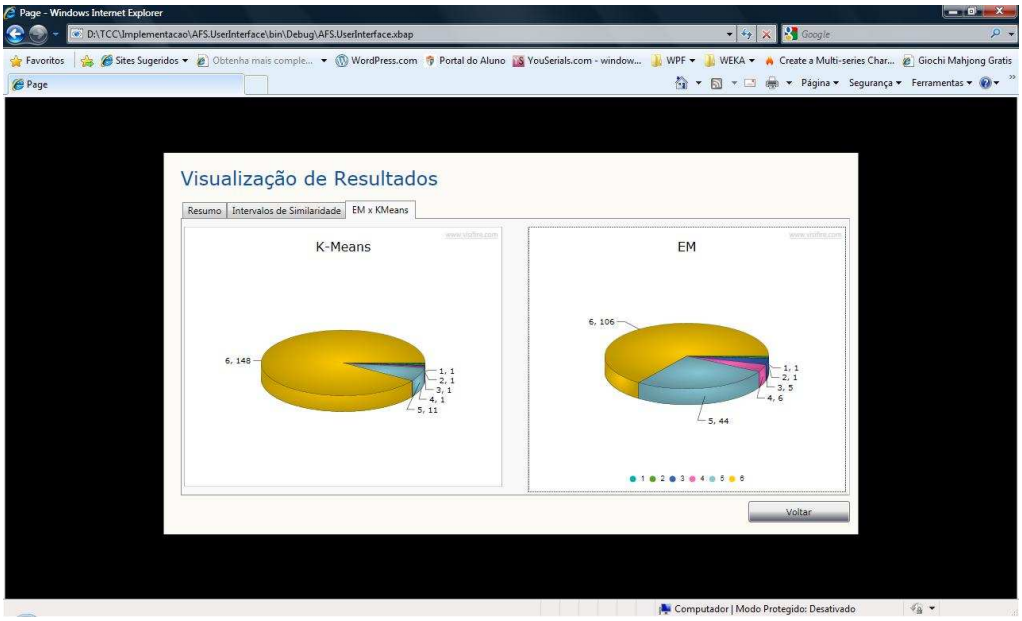

Figura 32 - Visualização de Resultados (Comparativo Em x K-Means)

A aba "EM x *K-Means*" exibida na Figura 32 ilustra da distribuição de documentos entre *clusters*, levando em consideração mesmo número *clusters*. O número de *clusters* em questão é igual ao número de clusters encontrados pelo algoritmo de EM durante a fase de análise.