UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

REDUÇÃO DE DIMENSIONALIDADE PARA AGRUPAMENTO DE TEXTO

TRABALHO DE GRADUAÇÃO

Aluno: Igor Cavalcanti Ramos (icr2@cin.ufpe.br)
Orientadora: Flávia de Almeida Barros (fab@cin.ufpe.br)
Co-orientador: Ricardo Bastos C. Prudêncio (prudencio.ricardo@gmail.com)

24 de Janeiro de 2008

"Ciência da Computação tem tanto a ver com o computador como a Astronomia com o telescópio, a Biologia com o microscópio, ou a Química com os tubos de ensaio. A Ciência não estuda ferramentas, mas o que fazemos e o que descobrimos com elas".

Edsger Dijkstra

Agradecimentos

Por tudo que consegui até hoje, agradeço aos meus pais. Painho e mainha, amo vocês.

Agradeço a todos os professores que estiveram presentes durante toda a minha formação escolar, desde o Jardim de Infância até a vida acadêmica. Sem vocês eu não iria até onde quis ir.

Aos Professores Flávia e Ricardo, todo o meu carinho e respeito como agradecimento ao seu amplo apoio, durante a realização desse trabalho.

Sumário

Índice de Figuras

1. Introdução

Técnicas de Agrupamento têm como função a organização de um conjunto de objetos em coleções que contêm apenas objetos similares [11]. Este tipo de técnica é recomendado quando os objetos em questão não estão previamente discriminados em classes, não havendo, assim, a possibilidade de separação manual desses objetos. Essas técnicas são úteis em algumas tarefas da área de Recuperação de Informação, como agrupamento automático de textos não previamente etiquetados. Também podem ser utilizadas para os mais diversos propósitos, como agrupamento de espécies de plantas e animais [6], assim como para indexação de imagens e perfil de consumidores [7].

No agrupamento de documentos, o fato de que esses são quase sempre representados por uma grande quantidade de palavras cria um problema de dimensionalidade. A dimensionalidade do vetor de termos dos documentos (no Modelo de Espaço Vetorial) afeta direta e negativamente a performance dos algoritmos de Agrupamento [1]. Assim sendo, é de grande importância a existência de técnicas para reduzir essa dimensionalidade.

O objetivo deste Trabalho de Graduação foi analisar alguns algoritmos de redução de dimensionalidade (seleção de características) de documentos para agrupamento de texto. O Capítulo 2 apresenta tipos de algoritmos de Agrupamento; no Capítulo 3 é feita uma explicação mais elaborada sobre os algoritmos de seleção de características; O Capítulo 4 traz uma descrição detalhada sobre o trabalho realizado, seguido pela apresentação dos resultados obtidos, com uma visão comparativa entre as técnicas implementadas. Por fim, o Capítulo 6 traz as conclusões desse Trabalho de Graduação.

2. Agrupamento de Documentos

A técnica de Agrupamento de Documentos (*Document Clustering*) foi inicialmente proposta para melhorar a precisão dos sistemas de Recuperação de Informação (RI), como um caminho eficiente de encontrar documentos similares [5] e como ferramenta para reconhecimento de padrões [9]. Este tipo de técnica é recomendado quando não há uma discriminação prévia de classes, sendo útil em casos onde não há a possibilidade da realização de separação manual de objetos em classes.

A execução de agrupamento automático de documentos baseia-se na hipótese de que documentos semelhantes tendem a permanecer em um mesmo grupo (cluster), pois possuem atributos em comum (*Cluster Hypothesis*). Duas importantes abordagens são: *Hierarchical Clustering* e *Partitional Clustering* [11].

A primeira abordagem é tida como tecnicamente melhor para criação de agrupamentos, mas na prática, é limitada pelo fato de ter complexidade quadrática de tempo. Essa abordagem é dividida em dois tipos: *Agglomerative* e *Divisive*. Na abordagem aglomerativa ("*bottom-up*"), os agrupamentos são criados a partir criação de uma hierarquia de agrupamentos; o algoritmo começa com documentos individuais que são continuamente agrupados, baseando-se numa definição de similaridade ou distância entre os mesmos. A abordagem divisiva faz o caminho inverso ("*top-down*").

Ao final, uma estrutura similar a uma árvore é obtida, como pode ser visto na figura 2.1 a seguir:

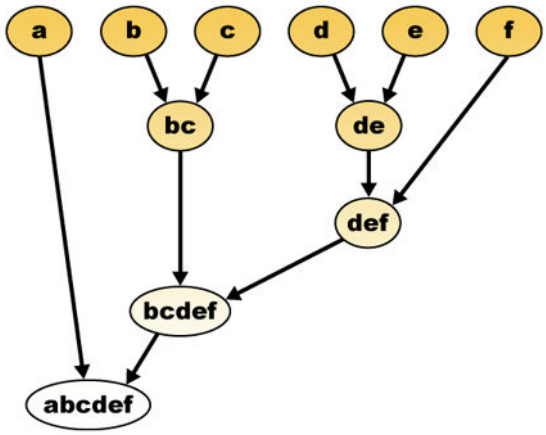

Figura 2.1Árvore criada por algoritmos hierárquicos [14].

A abordagem Particional de agrupamento, é caracterizada pela partição linear dos documentos, constituindo grupos disjuntos, distintos e não hierárquicos. Essa abordagem não cria agrupamentos antecessores ao resultado final. Se K agrupamentos são desejados, esses são encontrados todos de uma única vez, e durante a execução dos algoritmos os agrupamentos são apenas aperfeiçoados. A abordagem Particional possui complexidade linear de execução com relação ao número de documentos [5].

Nesse Trabalho de Graduação, escolhemos trabalhar com a abordagem Particional, uma vez que nosso objetivo principal era a análise de técnicas se seleção de características. Utilizamos o algoritmo *K-Means* [5] [6] [7] [9], por ser consideravelmente simples de implementar e ter baixo custo de execução. Esse fato possibilita, por exemplo, execuções repetidas para a seleção dos melhores resultados obtidos [8].

A seção 2.1, a seguir, apresenta algumas noções sobre o Modelo de Espaço Vetorial.

2.1 O Modelo de Espaço Vetorial

Para o algoritmo de agrupamento que foi utilizado nesse trabalho, os documentos são representados usando o Modelo de Espaço Vetorial. Nesse modelo, cada documento é considerado como um vetor, ***d***, no espaço de termos; as dimensões desse vetor são as palavras do mesmo e o valor de cada dimensão (o peso do termo) é dado por *Term Frequency*, que é a freqüência de um termo no documento. A figura 2.2 a seguir mostra as fases da construção da representação de um documento:

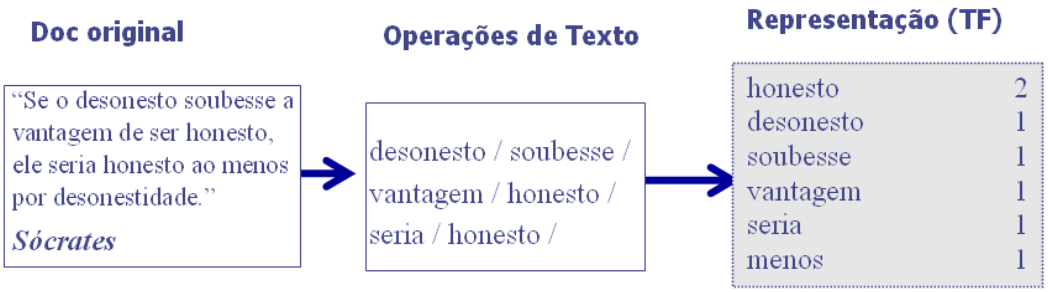

Figura 2.2 Criação da Representação de um documento.

De forma simplificada, cada vetor de documento é dado pela equação (1):

$$\boldsymbol{d}_{tf} = (tf_1, tf_2, \ldots, tf_n) \qquad \textbf{Eq. (1)}$$

Onde tf_i é a freqüência do i-ésimo termo no documento. Normalmente na fase que antecede a Representação, são removidas palavras sem significado completo (artigos, preposições, etc.), ou as palavras são reduzidas a sua forma canônica ("crescer", "crescimento", "decrescimento" = "cresc"), etc.

A figura a seguir mostra a representação de três documentos num Modelo de Espaço Vetorial (MEV) formado por três dimensões (Figura 2.3):

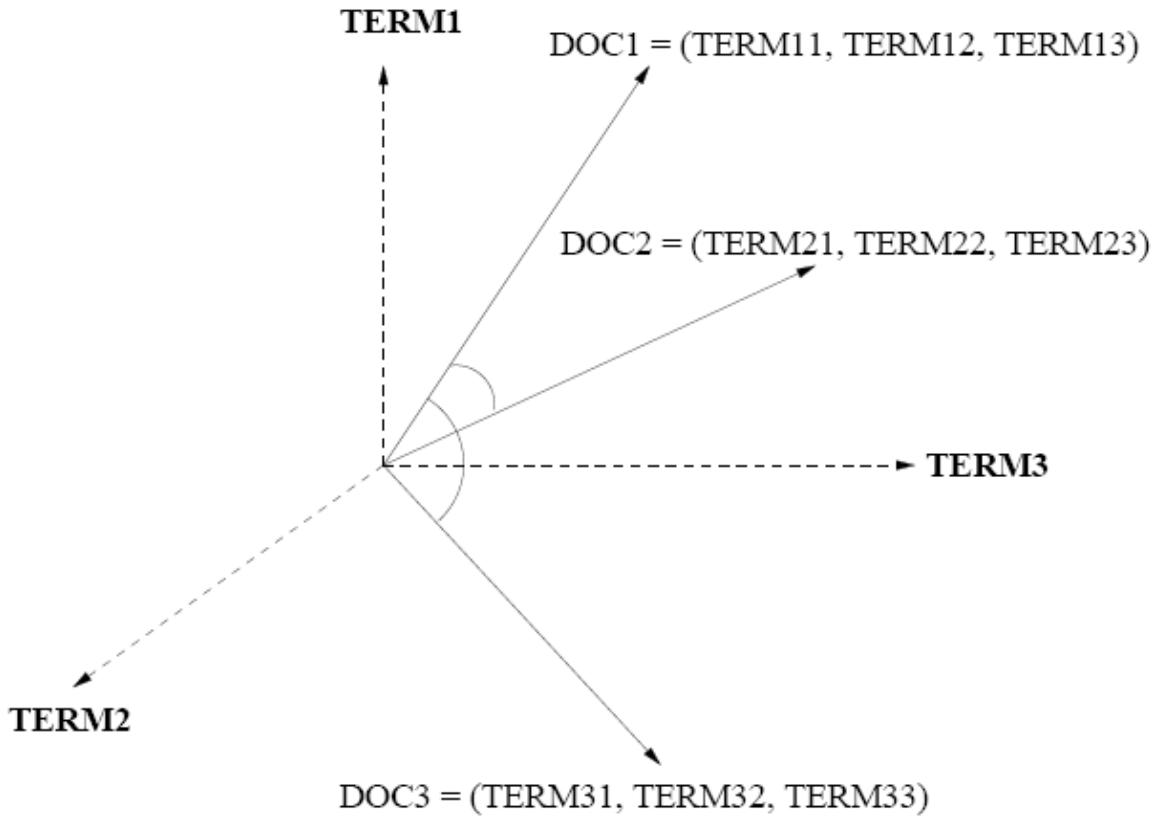

Figura 2.3 Modelo de Espaço Vetorial.

Note para que esse tipo de representação possa ser construído, todos os vetores dos documentos devem ter a mesma quantidade de dimensões, ou seja, os vetores precisam utilizar o mesmo MEV.

Nesse Trabalho todos os documentos são inicialmente lidos, preparados e sua representação inicial é utilizada apenas para a criação do MVE. Após essa criação, todos os documentos são "levados" a esse espaço, ou seja, todos os vetores dos documentos terão todas as dimensões do MVE criado.

Para esse trabalho é importante entender a noção de **centróide**. Dado um conjunto *S*, de documentos e suas respectivas representações vetoriais, o centróide é definido como na equação (2):

$$\mathbf{c} = \frac{1}{|S|} \sum_{d \in S} \mathbf{d} \qquad \textbf{Eq. (2)}$$

O centróide é apenas um vetor obtido pela média das freqüências dos termos, dos documentos presentes no conjunto *S*. O *K-Means*, como veremos adiante, utiliza a idéia do centróide, como a representação de todos os documentos de um agrupamento. A partir desse vetor, todos os vetores dos documentos são avaliados, no intuito de se descobrir qual é o agrupamento de um documento qualquer.

Outro importante conceito para o trabalho é Distância Euclidiana. A Distância Euclidiana é utilizada para o cálculo da distância entre dois vetores. A fórmula da distância é dada pela equação (3):

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots} \qquad \textbf{Eq. (3)}$$

Onde *a* e *b* são vetores.

2.2 Algoritmo K-Means

Entre as soluções de agrupamento, talvez a mais largamente utilizada e estudada seja o *K-Means* [5], [9]. O *K-Means* é baseado na idéia de que um ponto central pode representar um agrupamento.

O objetivo desse algoritmo é a redução da variância interna de cada um dos agrupamentos, também chamada de função de erro quadrática (equação 4), ou seja, o algoritmo busca diminuir as diferenças de todos os documentos em relação ao centróide de cada agrupamento [8].

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2 \qquad \textbf{Eq. (4)}$$

Onde K, é o número de agrupamentos S, onde i = 1, 2,..., k e μi é o centróide originado de todos os documentos $x_j \in S_i$.

Os passos do K-Means básico para encontrar K agrupamentos são:

1. Selecione K vetores (de documentos) no Modelo de Espaço Vetorial, como centróides iniciais.
2. Atribua todos os pontos para o centróide mais próximo.
3. Recompute os centróides de cada um dos agrupamentos. Isso atualiza a representação do agrupamento.
4. Repita os passos dois e três até que alguma condição de convergência tenha sido encontrada.

A figura abaixo mostra de forma resumida a execução do *K-Means*:

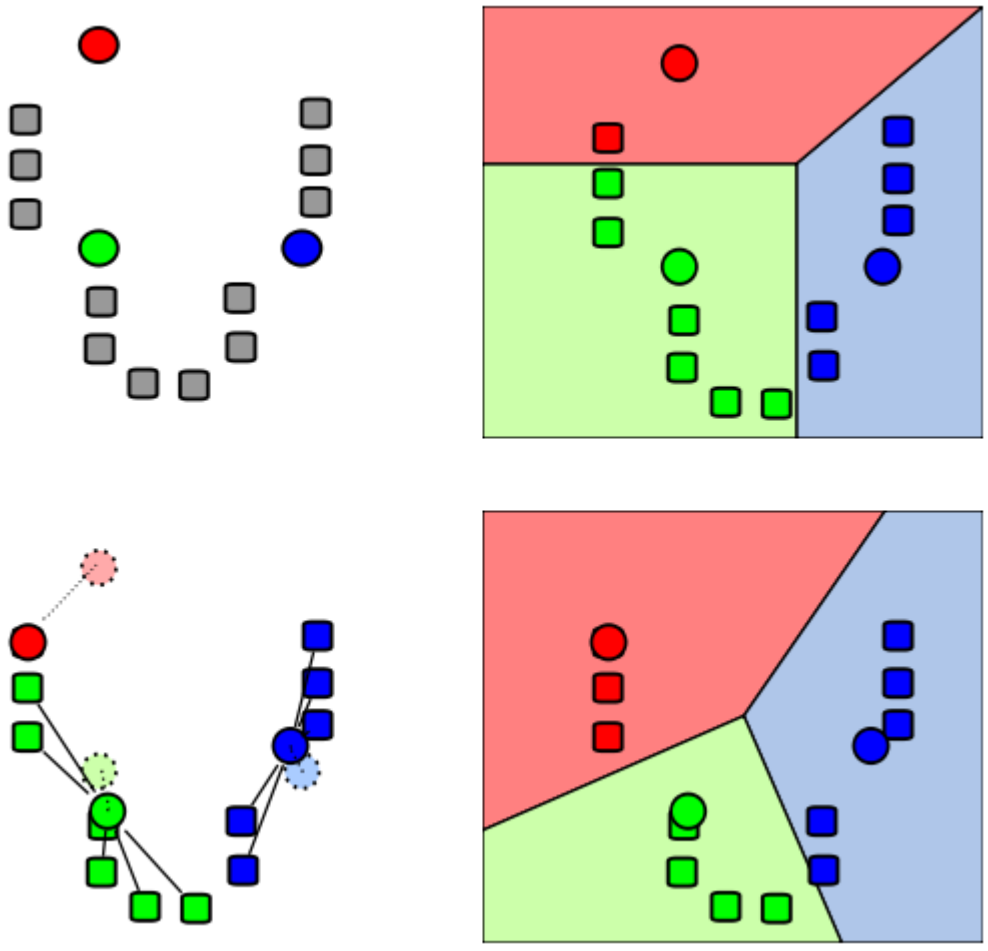

Figura 2.4 K-Means em execução [8].

A seleção inicial básica dos centróides de cada um dos clusters é feita através da escolha aleatória de vetores documentos, mas esse tipo de abordagem pode aumentar o tempo de execução do algoritmo, isso porque os resultados gerados pelo *K-Means* são fortemente influenciados pela escolha dos centróides iniciais. Por exemplo, ao serem escolhidos dois vetores muito "próximos", o algoritmo irá demorar mais pra convergir. Para uma melhor performance essa escolha pode estar apoiada em alguma heurística, como veremos mais adiante.

A grande vantagem desse algoritmo é a sua simplicidade e velocidade de execução; um tradicional algoritmo hierárquico, o *Simple Agglomerative Clustering*, possui complexidade quadrática de execução no número de documentos da base. Este baixo custo computacional possibilita a utilização prática no agrupamento de grandes quantidades de documentos.

Uma grande desvantagem do *K-Means* é que por ser tratar de uma técnica de contexto não-supervisionado é necessário informar a quantidade de agrupamentos desejados, esse fato pode influenciar na qualidade dos agrupamentos encontrados pelo do algoritmo. [11].

2.3 Considerações Finais

Vimos nesse capítulo importantes noções sobre o que é agrupamento de texto, além de uma breve explicação sobre o tipo de representação dos documentos e vimos também, uma pequena introdução sobre um famoso algoritmo de agrupamento, o *K-Means*. O próximo capítulo irá abordar a seleção de características e algumas técnicas para a realização dessa tarefa.

3. Técnicas de seleção de atributos

Como visto os algoritmos de Agrupamento, trabalham com representações internas dos documentos (em geral, vetores de termos em um espaço n-dimensional) a serem agrupados.

Esses termos, usados para indexar os documentos, são chamados aqui de atributos, ou características do documento.

Devido à grande quantidade de termos usados para representar esses documentos, alguns algoritmos tornam-se ineficientes (com alto custo de execução e baixa precisão na formação de Agrupamentos). Nesse contexto, surge a necessidade de se reduzir a dimensão da representação dos vetores, sem, contudo perder características importantes, i.e., atributos que caracterizam bem o documento em questão.

As técnicas de seleção de características, também chamadas de Seleção de Variáveis ou Seleção de Subconjuntos [11], melhoram de forma significativa o agrupamento de documentos, uma vez que maximizam a utilização dos recursos computacionais, através da escolha das características mais significativas de cada documento [4]. Essas técnicas podem ser utilizadas em contextos de aprendizado supervisionado e não-supervisionado.

No contexto de aprendizado supervisionado, a base de documentos é dividida em classes por interferência humana. As classes pré-definidas são então utilizadas como base do conjunto de treinamento para posteriormente classificação de documentos em contexto não-supervisionado [2].

A Seleção não-supervisionada de características é baseada em heurísticas para estimar a qualidade dos termos que representam o conjunto de documentos. Para documentos textuais, que é o caso deste trabalho, as heurísticas têm como base a ocorrência de cada termo nos documentos da base. Esse tipo de abordagem economiza tempo da interferência humana, sendo ideal para tarefas de mineração de dados, já que sempre será necessário tratar uma enorme quantidade de documentos de vários tipos [2].

A seleção de características não é aplicada apenas à redução de dimensionalidade em documentos textuais. Ela também é particularmente importante em problemas da Biologia Molecular, que podem envolver milhares de características, e até mesmo em métodos de classificação de imagem, onde cada pixel é visto como uma característica [11].

A seguir, serão descritas algumas das técnicas de seleção para contextos não-supervisionados.

3.1 Document Frequency

Document Frequency (DF) é a quantidade de documentos, da base em questão, onde um termo ocorre. Esse valor é calculado para todos os termos (característica) presentes na representação da base de documentos [3], então um *ranking* com as características é criado. Uma característica será selecionada, a partir de sua classificação no *ranking* baseando-se num limiar numérico que indica a porcentagem de termos selecionados. Essa técnica é a mais simples e mais recomendada para grandes quantidades de dados, por ter custo aproximadamente linear com o número de documentos utilizados [1].

Ao utilizarmos DF, pressupomos que ou os termos não são informativos ou não influenciam a performance global da seleção. O grande problema da remoção de termos raros é que essa remoção não pode ser agressiva, pois um pressuposto largamente aceito em Recuperação de Informação é que termos raros são relativamente informativos. [3].

3.2 Term Variance Quality

Term Variance Quality (TVQ) foi inicialmente proposto por [4] como sendo equação (5):

$$q_0(\mathbf{t}) = \sum_{j=1}^{n_0} f_j^2 - \frac{1}{n_0}\left[\sum_{j=1}^{n_0} f_j\right]^2 \qquad \textbf{Eq. (5)}$$

Onde a freqüência de um termo *t* num documento *d* é *f* e onde n_0 é o total de documentos no conjunto de dados. Nota-se que $q_0(\mathbf{t})$ é proporcional à variância da freqüência do termo, ou seja, se a variação for pequena, o atributo não contém informação capaz de discriminar os documentos (ou melhor, discriminar grupos de documentos).

Da mesma forma que no DF, esse valor é calculado para todos os termos da base e então um *ranking* é criado. Uma característica será selecionada, a partir de sua classificação no *ranking,* baseando-se num limiar numérico que indica a porcentagem de termos a serem selecionados.

3.3 Term Strength

O *Term Strength* (TS) foi inicialmente idealizado para ser utilizado para redução de vocabulário em Recuperação de Informação [1]. O método estima a importância do termo baseando-se na co-ocorência de um termo em documentos que são semelhantes [3].

O TS baseia-se na heurística em que documentos com muitas palavras compartilhadas são relacionados, e que termos que estão presentes em documentos relacionados, são relativamente informativos. Por causa disso, essa abordagem possui duas etapas [2]:

- *Encontrar documentos similares*. Nessa fase, todos os documentos são comparados usando-se o cosseno do ângulo entre os seus vetores de características. Um limiar numérico definido previamente determina o valor a partir do qual os documentos são considerados semelhantes.

- *Calcular o Term Strength*. Nessa fase, o *s(t)* de um termo *t* é calculado através da probabilidade condicional do termo ocorrer num documento d_i dado que ele ocorreu num documento d_j, onde d_i e d_j são documentos semelhantes (equação 6).

$$s(t) = p(t \in d_i \mid t \in d_j), sim(d_i, d_j) > \xi \qquad \textbf{Eq. (6)}$$

Onde *sim (di, dj)* é dada pela equação (7),

$$\cos\theta = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \, \|\mathbf{v_2}\|} \qquad \textbf{Eq. (7)}$$

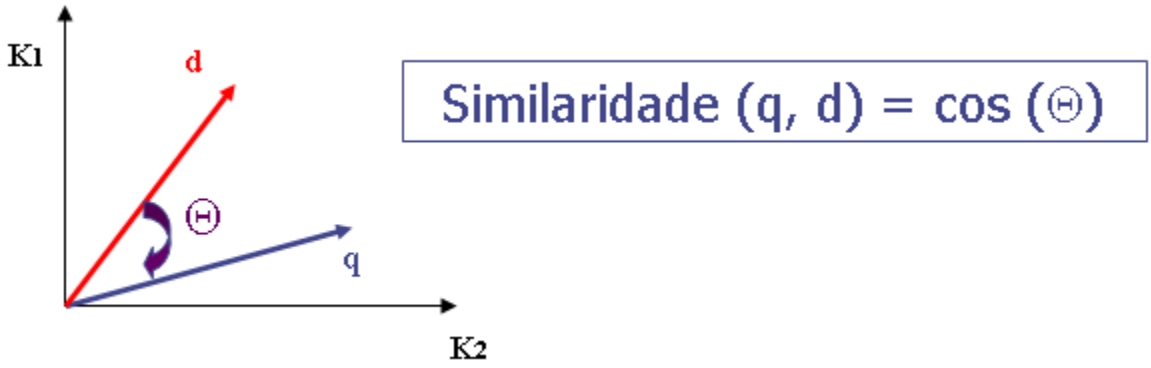

Figura 3.1 Similaridade entre vetores.

Uma característica será selecionada, a partir de sua classificação num *ranking* criado a partir dos resultados encontrados, baseando-se num limiar numérico que indica a porcentagem de termos a serem selecionados.

O grande problema desse algoritmo é a sua complexidade quadrática de execução com relação ao número de documentos da base de documentos [1].

3.4 Entropy-based Ranking

Nesse método, a força de um termo é medida pela diminuição da entropia obtida pela sua remoção. A equação da entropia é dada por equação (8):

$$E(t) = -\sum_{i=1}^{N}\sum_{j=1}^{N}(S_{i,j} \times \log(S_{i,j}) + (1 - S_{i,j}) \times \log(1 - S_{i,j})) \quad \textbf{Eq. (8)}$$

Onde $S_{i,j}$ é a similaridade entre os documentos d_i e d_j equação (9):

$$S_{i,j} = e^{-\alpha \times dist_{i,j}}, \alpha = -\frac{\ln(0.5)}{\overline{dist}} \quad \textbf{Eq. (9)}$$

Onde $dist_{i,j}$ é a distância entre os documentos d_i e d_j após a remoção do termo *t*, $\overline{dist}$ é a distância euclidiana média entre os documentos após a remoção do termo [1].

A Entropia é baixa em duas situações: quando a similaridade entre dois documentos é muito alta ou quando a similaridade é muito baixa. Se a Entropia é muito reduzida com a eliminação de um termo, isso significa que uma das duas situações será observada; considerando os termos remanescentes nos documentos; e nenhuma dessas situações é boa para a realização de agrupamento. O problema mais sério com esse método é a alta complexidade (***O (MN)^2***). Por esse motivo, é praticamente impossível a sua utilização em um conjunto grande de documentos, como no contexto para o qual esse trabalho é direcionado.

3.5 Term Contribution

O *Document Frequency* assume que os termos têm a mesma importância em documentos diferentes. Isso é um problema para os termos que podem aparecer em muitos documentos, mas são uniformemente distribuídos em diferentes classes, ou seja, que são pouco descritivos.

Segundo [1], a contribuição de um termo é a sua contribuição para a similaridade entre documentos. O resultado de agrupamentos de texto está fortemente ligado à similaridade dos documentos. A similaridade entre os documentos de dada pela equação (10):

$$sim(d_i, d_j) = \sum_t f(t, d_i) \times f(t, d_j)$$ **Eq. (10)**

Onde f(t, d) representa o tf * idf do termo t no documento d. Segundo [1] a contribuição do termo em um conjunto de dados é a contribuição total para os documentos, dada pela equação (11):

$$TC(t) = \sum_{i,j \cap i \neq j} f(t, d_i) \times f(t, d_j)$$ **Eq. (11)**

Então, o TC selecionará características que aparecem em poucos documentos ao invés de escolher aquelas que aparecem na maioria, ou seja, esse método tenta ignorar os termos mais freqüentes (e raros).

Sua principal desvantagem é a sua incapacidade de escolher termos que são, ao mesmo tempo, representativos e freqüentes.

3.6 Considerações Finais

Das cinco técnicas descritas, somente três foram utilizadas nesse trabalho: *Document Frequency, Term Variance Quality e Term Strength*. Essas técnicas foram escolhidas pelo fato de serem facilmente encontradas em trabalhos acadêmicos. Esse fato sugere que essas técnicas são populares e que possuem resultados realmente práticos.

O próximo capítulo apresenta algumas características práticas do trabalho desenvolvido. Nele vamos abordar desde a preparação da base de documentos à representação lógica dos mesmos.

4. Seleção de características para Clustering de texto

Como dito, este TG teve como objetivo central a análise de algoritmos de seleção de características para agrupamento de texto. Aqui, o algoritmo *K-Means* utilizado em combinação com três Selecionadores de Características: *Document Frequency* (DF), *Term Strength* (TS) e o *Term Variance Quality* (TVQ). O DF foi escolhido por sua simplicidade e por sua relatada eficiência. Já o TS foi escolhido por sua complexidade de execução.

Todos os sistemas resultantes desse Trabalho de Graduação foram implementados utilizando a linguagem Java.

4.1 Etapas de desenvolvimento do trabalho

Essa seção apresenta as etapas de desenvolvimento do trabalho, juntamente com alguns detalhes técnicos dos procedimentos implementados.

4.1.1 Seleção da Base de documentos

O primeiro passo foi escolher a base de documentos, como está descrito em detalhes na seção 5.1. No passo seguinte, foi criada a representação dos documentos usando o Modelo de Espaço Vetorial.

Para a criação das representações usando esse Modelo, foi necessária a implementação de um programa que pudesse ler todos os documentos da base, removendo a pontuação do texto, e selecionando as palavras sem repetição e que não estavam na *Stoplist*. Como resultado, uma lista foi criada e armazenada na memória do sistema. Nessa lista estão todas as dimensões do Modelo de Espaço Vetorial, ou seja, ela contém todas as características inicialmente adquiridas da base de documentos.

A *Stoplist* é uma lista com palavras consideradas irrelevantes, ou que não têm significado semântico associado (e.g., como artigos, preposições, conjunções). Por não serem úteis para a formação dos agrupamentos, é

interessante remover tais palavras dos documentos, para um melhor aproveitamento dos recursos computacionais. Aqui, essa lista foi criada manualmente, podendo ser facilmente encontrada na WEB. No nosso caso, ela está em Inglês.

Nesse ponto, a lista, de palavras obtidas através da remoção da pontuação até a remoção das *Stopwords*, é utilizada como base para o objetivo de estudo desse trabalho. Os selecionadores de características utilizam essa lista, como base para a redução da dimensionalidade da mesma, gerando como saída um novo Modelo de Espaço Vetorial, formado apenas pelas palavras (características) selecionadas.

Na subseção a seguir é mostrado, de forma geral, como os documentos foram representados.

4.1.2 Criação da representação dos documentos

Antes de iniciar a execução dos algoritmos de seleção, os documentos e agrupamentos necessitavam de uma representação computacionalmente adequada ao sistema.

No o intuito de alcançar esse objetivo, os documentos foram representados através de vetores de características. Nesses vetores, cada característica do documento representa uma das dimensões do vetor e cada dessas dimensões tem como valor, a freqüência da respectiva característica no documento, ou seja, se a palavra “graduação” aparecer duas vezes num determinado documento, o vetor desse documento terá para a sua dimensão “graduação” o valor 2 (dois).

Como Java é uma linguagem que utiliza o paradigma da Orientação a Objetos, foram criados objetos Java chamados intuitivamente de *Mean* e *Cluster*, para representar, respectivamente, os documentos e os agrupamentos no sistema.

O objeto *Mean*, como mencionado, é a representação de um documento. Ele formado por um objeto *java.lang.Integer* que representa a classe do documento, com valores numéricos. No sistema esses valores vão de 1 (um) a 10 (dez). Este tipo numérico é confrontado com o resultado final para o agrupamento como dito na seção 5.1; o objeto *Mean* também possui um objeto que é o vetor que representa o documento (*java.util.TreeMap<String, Long>*). Esse objeto foi escolhido para a representação do vetor, por ser formado por uma palavra e por um valor inteiro. O valor inteiro guarda a freqüência do termo no documento, ou seja, armazena o tamanho da dimensão desse vetor no Espaço Vetorial.

O Objeto *Cluster* representa o agrupamento de objetos *Mean*, ou seja, representa os agrupamentos de documentos. Cada objeto *Cluster* também possui um *java.lang.Integer*, mas que nesse caso é utilizado como o identificador do mesmo; também possui um objeto *Mean*, para a representação do centróide do agrupamento e finalmente, por um Objeto *java.util.ArrayList<Mean>*, que é uma lista que armazena todos os objetos *Mean* que fazem parte do agrupamento.

A próxima subseção explica com cada um dos selecionadores foi implementado no sistema.

4.1.3 Seleção de Características

Na implementação do *Document Frequency*, o selecionador mais simples dos que formam esse trabalho, a representação dos documentos segundo o Modelo Espaço Vetorial, que foi inicialmente criado apenas com a remoção de acentos e palavras inúteis, é varrido e o número de documentos em que cada característica presente é contabilizado e um *ranking* é criado. Então, estes valores são filtrados por um limiar numérico, que indica qual a porcentagem do termos a serem selecionados, informado no momento da chamada do algoritmo. Todos os termos cuja classificação no *ranking* esteja no grupo informado pelo

Limiar são selecionados, ou seja, se o limiar informado for 40% (quarenta porcento), somente as características que estão entre os primeiros quarenta porcento, são selecionadas.

O *Term Variance Quality* tem a parte inicial parecida com a parte inicial do *Document Frequency*, o Modelo de Espaço Vetorial inicial é varrido e o número de documentos em que cada característica está presente é contabilizado. A execução é divida na parte que calcula a soma dos quadros das freqüências dos termos (i) e na parte que calcula o quadrado da soma dos mesmos (ii). Ao fim desses cálculos, a parte (ii) é divida pelo número de documentos da base e o resultado é subtraído de (i). Assim como no DF, os resultados finais encontrados dão origem a um ranking e então, um limiar numérico percentual, que também informado no início da execução, é utilizado para a seleção das características.

O valor do limiar utilizado no TVQ está compreendido entre 0 (zero) e 100 (cem).

Por fim, o *Term Strength*. Esse é o mais complexo dos algoritmos utilizados. Inicialmente e de forma diferente dos algoritmos anteriores, O *Term Strength* utiliza o Modelo de Espaço Vetorial inicial, para o calculo das similaridades entre os documentos da base. A sua execução começa com a leitura dos documentos, para a criação dos seus vetores. Então, esses documentos são comparados utilizando o cosseno do ângulo entre os vetores e os documentos similares são verificados e demarcados. Os documentos somente são classificados como similares se a similaridade encontrada, for maior do que um limiar numérico informado no início da execução.

Após a verificação de similaridade, etapa com maior custo computacional, somente os termos dos documentos similares, têm as suas probabilidades condicionadas calculadas. Após o cálculo das probabilidades, um *ranking* com os valores encontrados é criado. As características cujas probabilidades, são selecionadas por limiar numérico percentual, também informado no início da execução, são selecionadas.

O valor do limiar utilizado para a classificação de similaridade, está compreendido entre os valores 0.0 (zero) e 1.0 (um); o valor limiar utilizado no cálculo das probabilidades está compreendido entre 0 (zero) e 100 (cem).

A figura 4.1 mostra um diagrama de classe das implementações dos selecionadores:

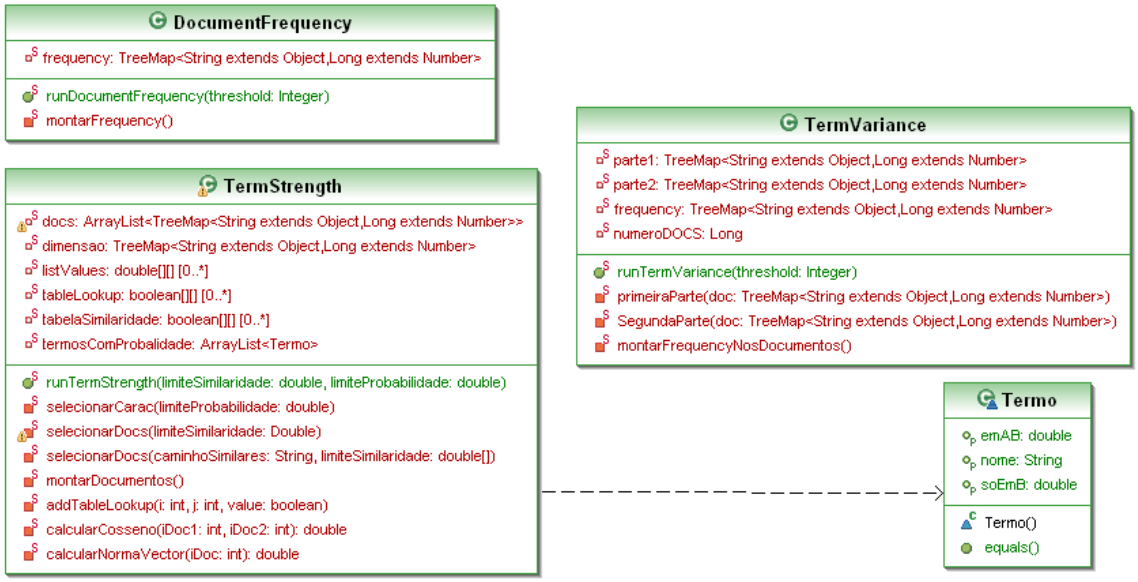

Figura 4.1 Diagrama de Classe dos Selecionadores de Características.

A seguir a explanação sobre a implementação do *K-Means*, seguida de um digrama de classes do sistema.

4.1.4 Agrupamento dos documentos

Como dito anteriormente, o resultado de todos os selecionadores é um novo Modelo de Espaço Vetorial formado pelas características selecionadas.

Durante o trabalho, os selecionadores foram usados de forma alternada, de modo que pude analisar o processo de Seleção de Características como veremos nos resultados posteriormente.

Após a seleção das características e a criação do Modelo, o passo seguinte deu início à implementação do *K-Means*.

Como dito, o *K-Means* utiliza os vetores dos documentos para a realização dos agrupamentos e necessita da informação prévia do número desejado de agrupamentos a serem construídos.

Para criar cada um dos objetos *Mean*, que representa os documentos, todos os seus vetores foram “levados” para o Modelo de Espaço Vetorial, ou seja, cada um dos documentos foi lido e apenas as características que estavam presentes no Modelo, foram consideradas relevantes e, portanto, fizeram parte do vetor que representou o documento.

Após a criação desses objetos, foi à vez dos objetos *Cluster*. Baseando-se na quantidade desejada de agrupamentos, que é informada antes do início da execução do algoritmo, os objetos *Cluster* foram criados e cada um recebeu um identificador. Depois que todos os objetos foram criados, os centróides foram devidamente escolhidos.

A escolha dos centróides iniciais é relatada na literatura como um fato crítico para o resultado do *K-Means*, devido ao fato dele apoiar-se na idéia de que um agrupamento pode ser representado por um “ponto” central.

A escolha, como dito na seção 2.2, pode aleatória no conjunto de objetos *Mean*, ou pode também ser aplicada através de heurísticas. Nesse Trabalho optei por uma heurística que verifica a similaridade baseada no cosseno do ângulo formado entre todos os centróides. Essa similaridade teria que ser igual a 0 (zero), indicando que os centróides são diferentes entre si.

Ao fim da seleção dos centróides, todos os objetos *Mean* tiveram calculadas as distâncias para esses centróides. Esse é o item 2 (dois) da seção 2.2. Um objeto *Mean* será adicionado ao *Cluster* para o qual a menor distância até o centróide do agrupamento, for encontrada.

A cada novo objeto adicionado a um *Cluster*, o cálculo do pré-processamento do próximo centróide é refeito, evitando assim, que o cálculo seja feito por completo, no momento em que o centróide do cluster deve ser substituído. Assim sendo, é fácil perceber que os centróides apenas são representados por vetores de documentos, no inicio da execução do algoritmo.

A versão utilizada do *K-Means* nesse Trabalho de Graduação, realiza o cálculo da Distância Euclidiana para calcular as distâncias entre vetores, que nesse caso é a distância entre os centróides dos agrupamentos e os documentos da base.

Ademais o algoritmo roda como descrito na seção 2.2 até que haja convergência, ou seja, quando os centróides não mais mudam.

A figura 4.2 mostra o diagrama de classe do *K-Means*:

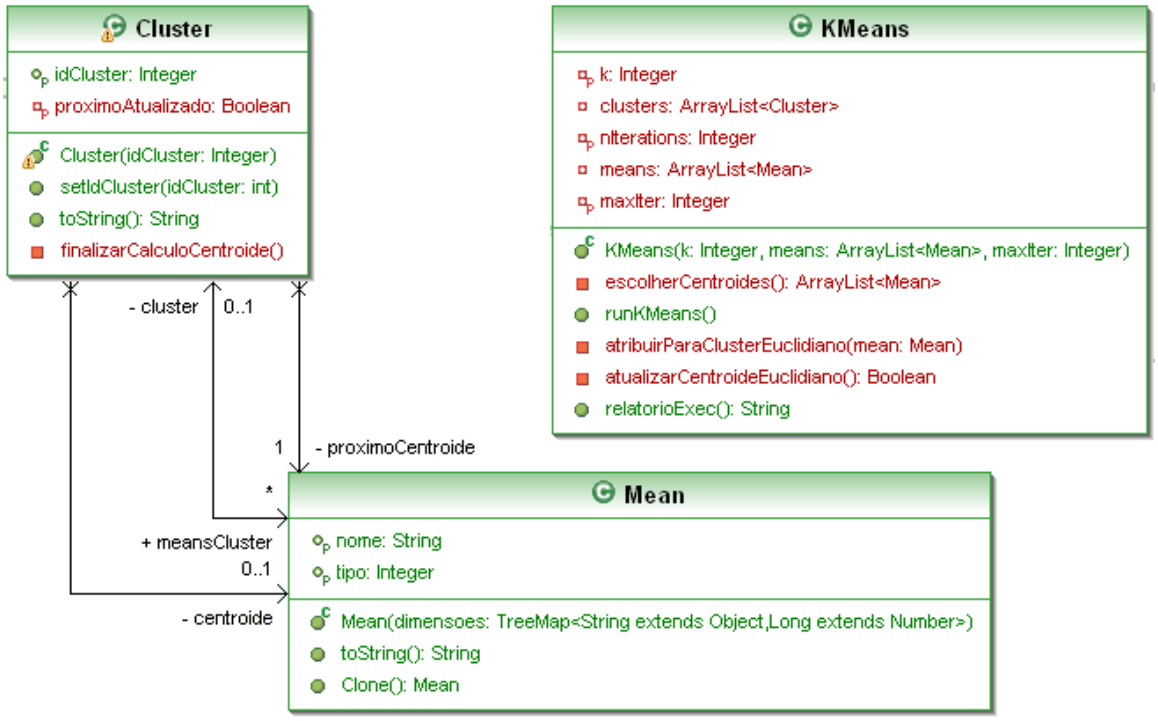

Figura 4.2 Diagrama de Classe do K-Means.

4.2 Considerações Finais

Nesse capítulo foram mostrados detalhes da implementação necessária para a realização das análises dos algoritmos de seleção. Fica claro que a implementação do algoritmo de seleção do K-Means é bastante simples.

No próximo capítulo será mostrado a metodologia de teste e os resultados obtidos a partir da execução do que foi descrito nesse capítulo.

5. Testes e resultados

Este capítulo apresenta detalhes sobre o corpus utilizado na realização deste trabalho, bem como a metodologia de testes e os resultados obtidos. O algoritmo *K-Means* implementado aqui foi executado com e sem os algoritmos de Seleção de Características, e a partir desses dados, gráficos foram criados com os resultados obtidos, dando uma visão mais completa dos experimentos.

5.1 Corpus

O corpus foi selecionado com a ajuda dos Orientadores, com base na idéia de analisar uma aplicação para WEB. Por essa razão, uma base do *dmoz* [12], formada por *WEB Snipets*, foi escolhida.

Os *WEB Snipets* são pequenos documentos constituídos com o título e uma pequena descrição do documento original. Eles são muito parecidos com os documentos retornados pelas consultas feitas a Engenhos de Busca.

Também era necessário que a base de documentos estivesse classificada, com o objetivo de facilitar a observação dos resultados gerados, por isso a base estava formada por 10 classes de documentos, num total de 1051 documentos. Essa classificação foi, segundo a fonte, realizada por pessoas, o que dá respaldo a um confronto dos resultados obtidos com essa classificação. As classes foram: *Agents (7,63%), Belief_networks (2,97%), Fuzzy (4,66%), Genetic (4,75%), Machine_learning (24,48%), Natural_language (12,38%), Neural_networks (29,13%), Philosophy (4,26%), Support_vector (2,77%)* e *Vision (6,93%)*.

5.2 Metodologia de teste

Nos experimentos realizados, o algoritmo *K-Means* foi executado para realizar o agrupamento de documentos com e sem seleção de características.

Como dito, as técnicas se seleção foram a *Document Frequency*, *Term Strength* e *Term Variance Quality*. Para cada técnica de seleção de características, avaliamos diferentes limiares que indicam a redução dimensional dos documentos.

Três medidas avaliam a qualidade dos grupos gerados nos experimentos no trabalho: Precisão, Cobertura e *F-Measure*.

A Precisão de um agrupamento é medida através da fração entre o número de documentos da classe de maior ocorrência no agrupamento e o número total de documentos do mesmo, por exemplo: Se em agrupamento a classe mais freqüente possui 40 (quarenta) documentos e o agrupamento possui 100 documentos, a precisão é de 0.4 (zero ponto quatro). Essa medida informa o quão "puro" o agrupamento é.

A Cobertura em um agrupamento é medida através da fração formada pelo número de documentos da classe de maior ocorrência no agrupamento, divido pelo número de total de documentos dessa classe, que estão presentes na base de documentos. Essa medida informa o quão eficiente, i.e., em agrupar termos de uma mesma classe, foi o *K-Means*, tendo em vista a classe do documento. No exemplo anterior, por exemplo, se o total de documentos da classe mais numeroso na base foi 60 (sessenta), então a Cobertura é 0.66 (zero ponto sessenta e seis).

Por fim, A *F-Measure* é a uma média Harmônica ponderada da Precisão e Cobertura (Equação 12):

$$F = 2 * (Precisão * Cobertura) / (Precisão + Cobertura) \qquad \textbf{Eq. (12)}$$

Como o *K-Means* gera vários agrupamentos e pelo fato desses poderem ser diferentes entre suas execuções, é necessário uma avaliação do resultado final como um todo.

Devido a isso, utilizei a *F-Measure* para todos os agrupamentos gerados. Para tal, fiz a média aritmética da Precisão e da Cobertura de todos os

agrupamentos e então utilizei os valores para o cálculo da *F-Measure*. Quanto maior for a *F-Measure*, melhor terá sido o resultado final.

Como a qualidade do *K-Means* depende dos centróides iniciais, executamos o algoritmo 07 (sete) vezes para cada limiar numérico e então, computamos o valor médio de *F-Measure*. No total cada um dos algoritmos de seleção, juntamente o *K-Means*, foram executados 42 (quarenta e duas) vezes.

Os resultados obtidos são mostrados na seção a seguir.

5.3 Resultados obtidos

Confuision Matriz é citada em [4]. Essa tabela é formada pelos resultados dos agrupamentos pelos tipos de cada documento. A seguir alguns exemplos dos resultados obtidos para a execução do K-Means sem selecionadores de características:

Cluster 0	1	8	1	1	106	3	74	2	8	0	TOTAL = 204	Prec: 0.51960784	Cob:0.43089432
Cluster 1	0	0	0	0	2	1	6	2	0	25	TOTAL = 36	Prec: 0.6944444	Cob:0.36231884
Cluster 2	5	27	31	6	29	20	138	14	4	7	TOTAL = 281	Prec: 0.4911032	Cob:0.47098976
Cluster 3	0	0	1	0	0	2	1	0	0	0	TOTAL = 4	Prec: 0.5	Cob:0.016129032
Cluster 4	0	1	0	0	6	0	15	0	1	14	TOTAL = 37	Prec: 0.4054054	Cob:0.051194537
Cluster 5	1	2	2	4	29	1	12	1	1	6	TOTAL = 59	Prec: 0.4915254	Cob:0.11788618
Cluster 6	0	0	0	0	1	0	0	0	0	0	TOTAL = 1	Prec: 1.0	Cob:0.0040650405
Cluster 7	52	2	0	0	3	1	1	0	0	0	TOTAL = 59	Prec: 0.88135594	Cob:0.6753247
Cluster 8	18	19	12	55	70	96	46	23	13	17	TOTAL = 369	Prec: 0.2601626	Cob:0.7741935
Cluster 9	0	0	0	1	0	0	0	0	0	0	TOTAL = 1	Prec: 1.0	Cob:0.014925373

Número de Iterações: **4.**
Número de Documentos: **1051.**
Tempo de execução: **458 segundos.**

F-Measure: **0, 397714249**

Cluster 0	2	11	1	1	126	3	79	4	9	0	TOTAL = 236	Prec: 0.5338983	Cob:0.5121951
Cluster 1	10	0	2	1	7	1	83	0	1	2	TOTAL = 107	Prec: 0.7757009	Cob:0.28327644
Cluster 2	0	0	0	0	0	1	0	0	0	0	TOTAL = 1	Prec: 1.0	Cob:0.008064516
Cluster 3	0	5	2	0	4	0	0	0	0	0	TOTAL = 11	Prec: 0.45454547	Cob:0.084745765
Cluster 4	13	39	38	24	83	113	114	34	16	64	TOTAL = 538	Prec: 0.21189591	Cob:0.3890785
Cluster 5	0	0	1	37	2	0	0	0	0	0	TOTAL = 40	Prec: 0.925	Cob:0.5522388
Cluster 6	52	1	0	0	1	1	0	0	0	0	TOTAL = 55	Prec: 0.94545454	Cob:0.6753247
Cluster 7	0	2	0	4	15	1	10	0	1	3	TOTAL = 36	Prec: 0.41666666	Cob:0.06097561

Cluster 8	0	1	3	0	8	4	6	4	0	0	TOTAL = 26	Prec: 0.30769232	Cob:0.032520324
Cluster 9	0	0	0	0	0	0	1	0	0	0	TOTAL = 1	Prec: 1.0	Cob:0.0034129692

Número de Iterações: **4.**
Número de Documentos: **1051.**
Tempo de execução: **684 segundos.**

F-Measure: **0, 372764568**

Cluster 0	51	1	0	0	1	1	0	0	0	0	TOTAL = 54	Prec: 0.9444444	Cob:0.66233766
Cluster 1	0	15	0	0	2	0	14	0	0	0	TOTAL = 31	Prec: 0.48387095	Cob:0.2542373
Cluster 2	0	0	0	0	0	0	0	1	0	0	TOTAL = 1	Prec: 1.0	Cob:0.023809524
Cluster 3	0	0	0	0	1	0	0	0	0	0	TOTAL = 1	Prec: 1.0	Cob:0.0040650405
Cluster 4	2	10	1	1	129	3	80	3	9	0	TOTAL = 238	Prec: 0.5420168	Cob:0.5243902
Cluster 5	0	0	0	0	0	0	0	1	0	0	TOTAL = 1	Prec: 1.0	Cob:0.023809524
Cluster 6	4	2	6	3	9	5	21	0	5	6	TOTAL = 61	Prec: 0.3442623	Cob:0.07167236
Cluster 7	0	5	2	0	4	0	0	0	0	0	TOTAL = 11	Prec: 0.45454547	Cob:0.084745765
Cluster 8	19	23	37	59	79	100	159	35	12	59	TOTAL = 582	Prec: 0.27319586	Cob:0.54266214
Cluster 9	1	3	1	4	21	15	19	2	1	4	TOTAL = 71	Prec: 0.29577464	Cob:0.085365854

Número de Iterações: **5.**
Número de Documentos: **1051.**
Tempo de execução: **588 segundos.**

F-Measure: **0, 335046712**

Resultado final para a execução do *K-Means* sem os selecionadores de características:

0, 338519

Tempo médio de execução:

556,7 segundos.

A execução do algoritmo *K-Means* sem a seleção de atributos, obteve *F-Measure* média de 0,33 e um tempo médio de execução de 556,7 segundos.

Os gráficos que serão apresentados aqui, demonstram a entre relação tempo de execução com porcentagem de termos selecionados; também a relação entre o valores de F-Measure obtidos e a porcentagem dos termos selecionados.

Estes valores percentuais de termos selecionados, informam sobre a quantidade de termos que foram selecionados, tendo com base o conjunto de todos os termos da base de documentos.

O primeiro gráfico (figura 5.1) mostra o tempo de execução necessário para a execução do *K-Means*, quando o algoritmo DF foi utilizado para a seleção de características. O gráfico foi montado considerando as quantidades de termos que foram selecionados pelo algoritmo de seleção de características.

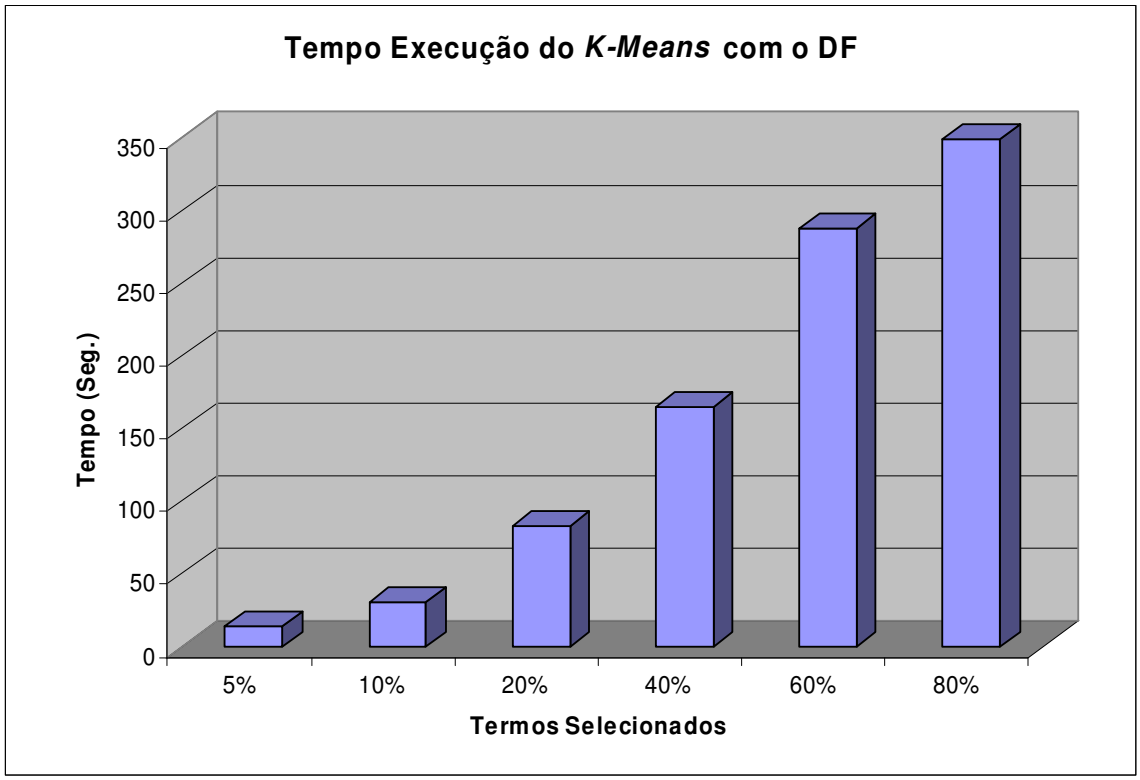

Figura 5.1 Gráfico do tempo de execução do K-Means com o DF.

Fica evidente que a seleção de características reduz o tempo de execução do algoritmo de agrupamento. Quando os 5% (cinco porcento) melhores termos são selecionados, o tempo de execução é de aproximadamente de 13 segundos e à medida que a quantidade de termos selecionados aumenta, o tempo de execução também aumenta.

O próximo gráfico (Figura 5.2), mostra os resultados obtidos para a *F-Measure* com relação à quantidade de termos selecionados.

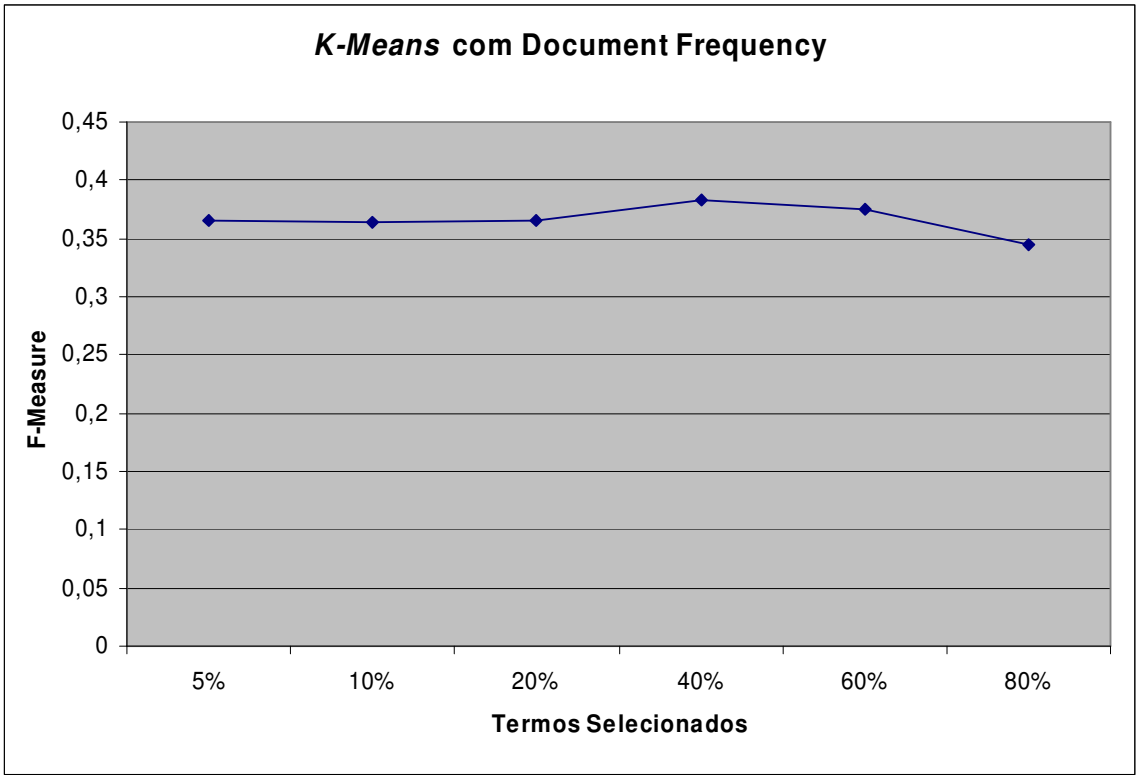

Figura 5.2 Gráfico com os valores da execução do K-Means com DF.

De modo geral, o *K-Means* teve quase o mesmo resultado durante os variados limiares de seleção de características do *Document Frequency*. Apenas quando os 40% (quarenta porcento) melhores termos foram escolhidos, os resultados tiveram uma pequena melhora. A partir desse valor, o *K-Means* começa a ter uma tendência de perda de rendimento, o que pode ser explicado pela alta quantidade de termos disponíveis para a representação dos documentos. Nesse experimento a *F-Measure* média foi 0,36 enquanto que o tempo médio de execução foi de 154 segundos.

Os gráficos seguintes (figuras 5.3 e 5.4) mostram os tempos de execução e os resultados baseados na *F-Measure* quando o selecionador TVQ é utilizado. Seguindo o mesmo raciocínio dos gráficos anteriores, foram escolhidos diferentes limiares indicando a quantidade de termos selecionados para a execução do *K-Means*.

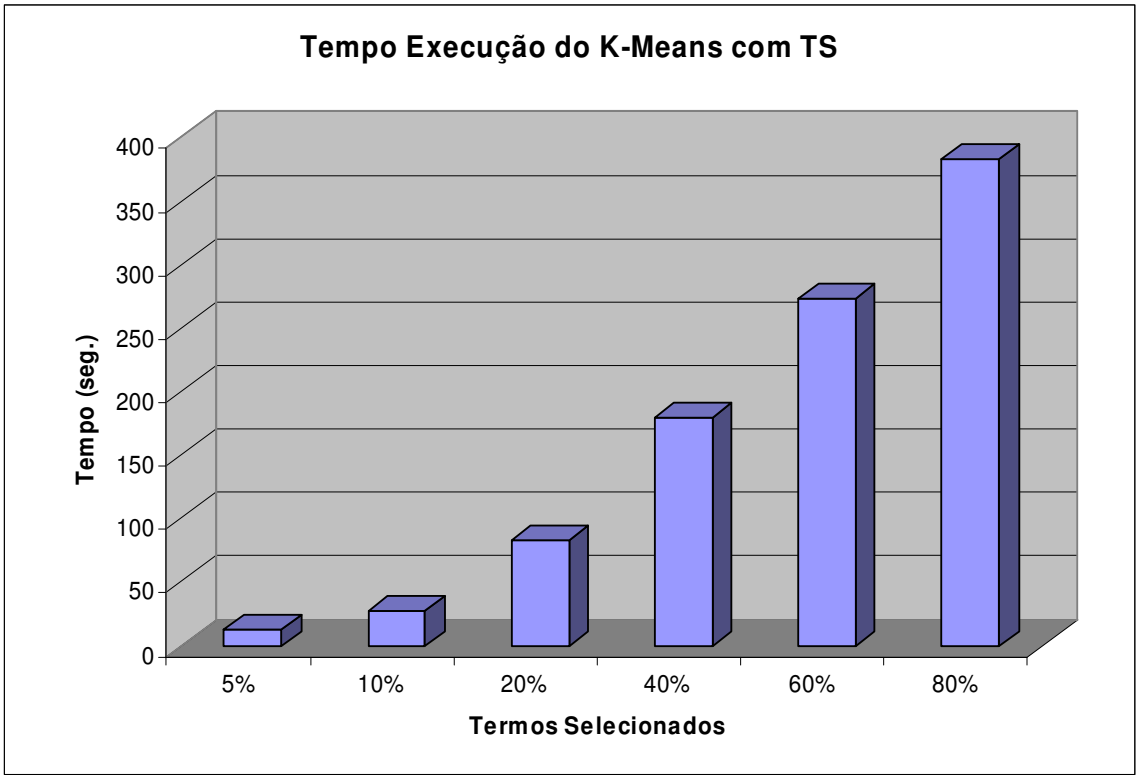

Figura 5.3 Gráfico do tempo de Execução do TVQ.

Como era esperado, o tempo de execução do *K-Means* foi drasticamente reduzido, com a utilização do algoritmo de seleção de características. O tempo de execução com 5% (cinco porcento) melhores termos foi, em média, apenas 12 segundos, enquanto que a execução do algoritmo com 80% (oitenta porcento) melhores termos, foi de 382 segundos em média, ou seja, muito a baixo do resultado obtido com a execução do *K-Means* sem selecionadores (média > 500 segundos).

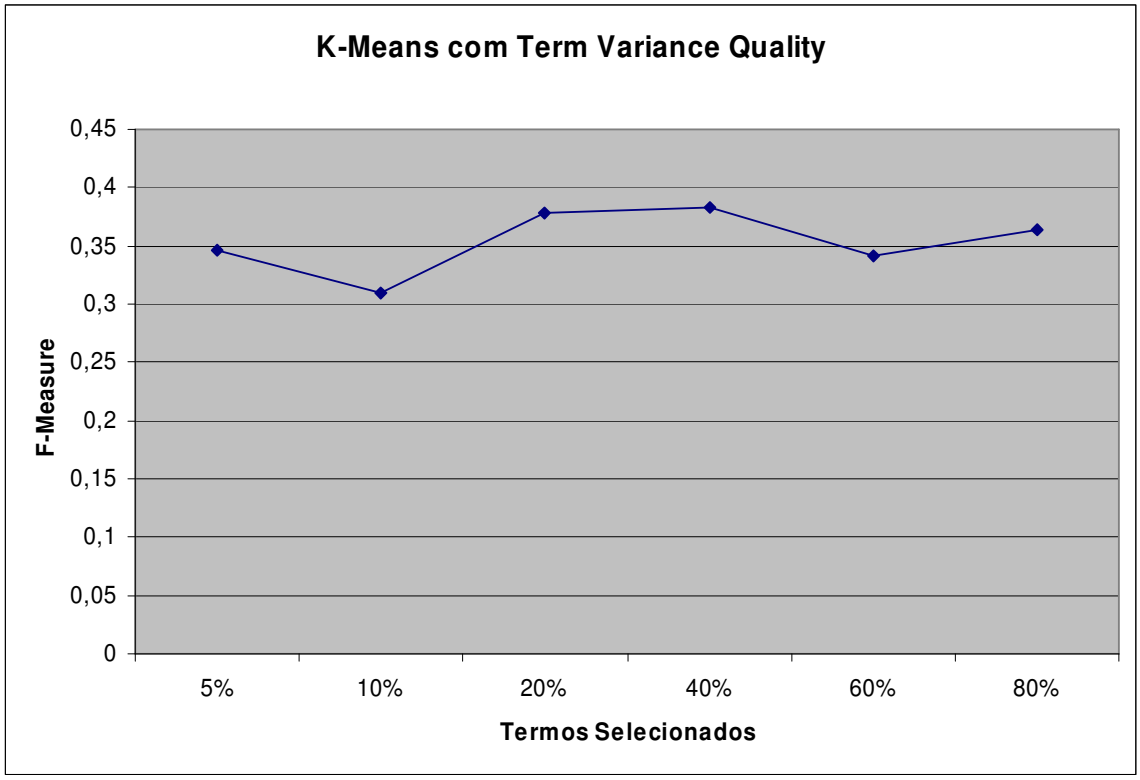

Figura 5.4 Gráfico com os resultados da execução do TVQ.

De forma mais evidente, o TVQ mostra variações no rendimento durante todo o experimento. Mesmo quando a quantidade de termos selecionados começa a ficar alta, os resultados parecem mudar pouco. Nesse experimento a *F-Measure* média foi de 0,35 e o tempo médio, mostrado na figura 5.3, foi de 159 segundos.

Por fim, o gráfico da execução do *K-Means* utilizando o Term Strength (figura 5.5). Como o TS necessita de dois parâmetros, um informa o limiar da similaridade entre os documentos e o outro informa sobre a porcentagem de características a serem selecionadas, adotei como limiar de similaridade um valor fixo de 60% (sessenta por cento). Esse valor, classifica como similares aproximadamente 20% (vinte por cento) dos documentos da base, ou seja, somente palavras de documentos muito similares serão observadas. Valores menores do que este comprometem seriamente os resultados, pois muitas características estavam, no momento do cálculo das suas probabilidades

condicionadas, recebendo 0 (zero) como probabilidade, por não estarem presentes em ambos os documentos.

Devido às suas características, o *Term Strength* foi o algoritmo que mais necessitou de tempo para a execução da seleção de características, devido à sua complexidade quadrática no número de documentos. O seu maior gasto está em conferir a similaridade entre dois documentos e por isso, o tempo médio final de execução do *K-Means* utilizando o TS é tão grande e desproporcional, em relação aos experimentos anteriores, que nem mesmo é útil compará-los. Os resultados encontrados estiveram em algo em torno de 3700 (três mil e setecentos) segundos por execução.

A figura seguinte mostra o gráfico do experimento para limiares percutais de seleção de características.

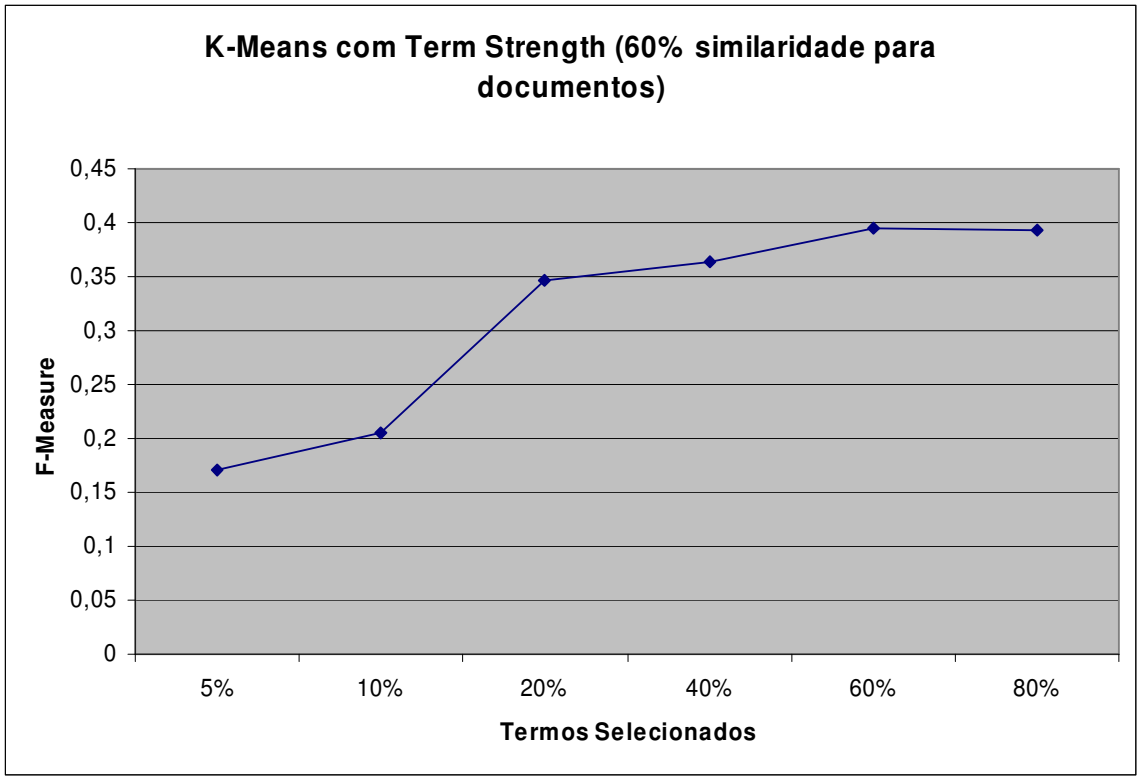

Figura 5.5 Gráfico com os resultados da execução do TS.

Diferentemente dos experimentos anteriores, a execução do *K-Means* utilizando o TS com selecionador de características, teve resultados muito diferentes com relação à quantidade de termos selecionados.

Vale lembrar que pelo fato dos documentos serem, num primeiro momento, pré-selecionados através do calculo da similaridade, diminui consideravelmente o número de característica disponíveis, antes mesmo da utilização do limiar percentual. A seleção de documentos similares gera um efeito de redução do número documentos da base e por conseqüência, o número de termos para a execução dos selecionadores.

A pré-seleção de documentos explica o porquê de o resultado ser tão ruim quando poucas características são selecionadas. Como a quantidade de termos inicialmente já é reduzida, selecionar apenas 5% (cinco porcento) desses termos acarreta na falta de características para a criação do Modelo Espaço Vetorial, ou seja, os vetores que representam os documentos são formados por pouquíssimas palavras. Com vetores pequenos, muitos documentos não podem ser representados no Modelo. Por essa razão o resultado foi ruim com percentuais pequenos.

Mas à medida que o número de termos foi aumentando os valores dos resultados obtiveram melhoras.

A média dos resultados da *F-Measure* encontrada aqui, foi de 0,31.

5.4 Considerações Finais

Nesse capítulo vimos os resultados obtidos pela a execução do *K-Means* utilizando os algoritmos de seleção de características implementados nesse Trabalho de Graduação. A forma como os resultados foram obtidos possibilitou uma visão analítica da performance dos algoritmos, com resultados comparativos e com visões específicas das execuções dos experimentos, alcançando portanto, o objetivo proposto nesse Trabalho.

6. Conclusão

Como pode ser observado nos experimentos, num ambiente de grande volume de dados, as técnicas de seleção de características DF e o TVQ reduzem drasticamente o tempo de execução do algoritmo de agrupamento *K-Means*. O TS, como relatado anteriormente, possui complexidade quadrática no número de documentos da base e esse fato praticamente inviabiliza a sua utilização em bases relativamente grandes, como a que foi utilizada nesse trabalho.

Com relação à medida *F-Measure,* todos os algoritmos de alguma forma melhoram os agrupamentos finais. Durante os experimentos, como por exemplo, o DF, mesmo utilizando altos percentuais de seleção de termos, há uma melhora nos resultados comparativamente com a execução do *K-Means* sem os selecionadores.

Os melhores resultados de *F-Measure* média, como pode ser visto através das figuras 5.2, 5.4 e 5.5, foram encontrados na execução do *K-Means* tendo como selecionador de características o TS.

Referências Bibliográficas

[1]	Liu, T.; Liu, S.; Chen, Z.; Ma, W.-Y.: An evaluation on feature selection for text clustering. In ICML '03, pp. 488-495, 2003. http://research.microsoft.com/~zhengc/papers/ICML2003-15.pdf
[2]	Do, T. Dung; Hui, S.Cheung; Fong, A.: Associative Feature Selection for Text Mining. In International Journal of Information Technology, Vol. 12 No.4. http://www.icis.ntu.edu.sg/scs-ijit/1204/1204_7.pdf
[3]	Pedersen, J. O.; Yang, Y.: A Comparative Study on Feature Selection in Text Categorization. In Proceedings of the Fourteenth International Conference on Machine Learning, Pages: 412 - 420 , ISBN:1-55860-486-3 , 1997. http://portal.acm.org/citation.cfm?id=657137
[4]	Nicholas, C.; Kogan, J.; Dhillon, I.: Feature Selection and Document Clustering. http://www.csee.umbc.edu/cadip/2002Symposium/kogan.pdf
[5]	M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In KDD Workshop on Text Mining, 2000. http://citeseer.ist.psu.edu/steinbach00comparison.html
[6]	Faber, V.: Clustering and the Continuos K-Means Algorithm. In Los Alamos Science, Number 22 ,1994. http://www.fas.org/sgp/othergov/doe/lanl/pubs/00412967.pdf
[7]	Modha, D. and Spangler, W. Feature Weighting in k-Means Clustering. Machine Learning, 47, 2002. http://citeseer.ist.psu.edu/modha02feature.html
[8]	K-means algorithm. In Wikipedia. http://en.wikipedia.org/wiki/K-means_algorithm
[9]	Kanungo, T., D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu (2002). An efficient k-means clustering algorithm: analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7): 881-892. http://citeseer.ist.psu.edu/kanungo00efficient.html
[10]	Centroid. http://en.wikipedia.org/wiki/Centroid
[11]	Law, M.H.; Jain, A.K.; Figueiredo, M.A.T: Feature selection in mixture-based clustering. In Advances in Neural Information Processing Systems, volume 15, 2003. http://www.cse.msu.edu/~lawhiu/papers/TPAMI-LawFiguredoJain.pdf
[12]	Base de Dados (*Corpus*) http://www.dmoz.org/
[13]	Dash, M.; Liu, H.: Feature Selection for Clustering. In Lecture Notes In Computer Science; Volume 1805, 2000. http://citeseer.ist.psu.edu/613077.html
[14]	Cluster Analysis. In Wikipedia. http://en.wikipedia.org/wiki/Data_clustering

ASSINATURAS

Este Trabalho de Graduação é resultado dos esforços do aluno Igor Cavalcanti Ramos, sob a orientação dos professores Flávia de Almeida Barros e Ricardo Prudêncio sob o título: “Redução de Dimensionalidade para Agrupamento de Texto”. Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste Trabalho de Graduação.

_____
Igor Cavalcanti Ramos

_____
Flávia de Almeida Barros

_____
Ricardo B. C. Prudêncio