FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

VINICIUS MURATA SAITO

Desenvolvimento de uma Ferramenta para a Categorização Automática de Produtos de um Marketplace

Marília
2014

VINICIUS MURATA SAITO

Desenvolvimento de uma Ferramenta para a Categorização Automática de Produtos de um Marketplace

Monografia apresentada ao Centro Universitário Eurípides de Marília como parte dos requisitos necessários para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Ms. Ricardo Sabatine

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Vinicius Murata Saito

Desenvolvimento de um Webservice para a Categorização Automática de Produtos de um Marketplace

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Sistemas de Informação.

Nota: 8,5 (oito e meio)

Orientador: Ricardo José Sabatine

1°. Examinador: Fabio Lucio Meira

2°. Examinador: Paulo Rogerio de Mello Cardoso

Marília, 02 de dezembro de 2014.

AGRADECIMENTOS

Ao Orientador Prof. Ms. Ricardo Sabatine pelo incentivo, simpatia e presteza no auxílio às atividades e discussões sobre o andamento e normatização desta Monografia de Conclusão de Curso.

A minha família por ter promovido a minha vida acadêmica.

E finalmente, a DEUS pela oportunidade e pelo privilégio que me foram dados em compartilhar tamanha experiência de, ao frequentar este curso, poder perceber e atentar para a relevância de temas que não faziam parte, em profundidade, das nossas vidas.

SAITO, Vinicius Murata. **Desenvolvimento de uma ferramenta para categorização automática de produtos de um marketplace**. 2014. 51 f. Trabalho de Curso (Bacharelado em Sistemas de Informação) – Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2014

RESUMO

Com a tendência do surgimento de marketplaces onde, há várias lojas virtuais vendendo seus produtos, ocorrem problemas no agrupamento desses produtos, já que cada lojista tende a categorizar seus produtos de forma diferenciada, provocando um problema para o marketplace no agrupamento dessas ofertas em suas categorias e assim, dificultando a navegação e a visualização de produtos, prejudicando o usuário que não consegue encontrar um determinado produto e o anunciante que poderá ter um produto não encontrado. Por isso, propõe-se nesse projeto o desenvolvimento de uma aplicação que seja capaz de agrupar e categorizar automaticamente essas ofertas com o uso de *frameworks* para o processamento distribuído escalável para o processamento das informações textuais, integrada a ferramentas de inteligência artificial para o agrupamento das informações textuais presentes em cada oferta.

Palavras-chaves: Categorização; aprendizado-máquina; classificação textual e agrupamento de dados.

SAITO, Vinicius Murata. **Desenvolvimento de uma ferramenta para categorização automática de produtos de um marketplace**. 2014. 51 f. Trabalho de Curso (Bacharelado em Sistemas de Informação) – Centro Universitário Eurípides de Marília, Fundação de Ensino "Eurípides Soares da Rocha", Marília, 2014

ABSTRACT

With the trend of the emergence of marketplace, where there are several online stores selling their products, problems occur in grouping the products, as each shopkeeper tends to categorize their products differently, causing a problem for the marketplace in the grouping of these offers in each category and making difficult to navigate and find the product displayed, harming to the user who cannot find a particular product and the advertiser that has a product not found. Therefore, it is proposed in this project to develop an application that is able to group and automatically categorize these offers using frameworks for scalable distributed processing, integrated with artificial intelligence library for group this textual information used in each offer.

Keywords: Categorization; machine learning; textual classification and clustering.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Introdução

O processo de manipulação de informações hoje está cada vez mais incorporado ao dia-a-dia das empresas e pessoas. Porém, esses grandes volumes de informações tornam-se cada vez mais difícil à assimilação da informação, renovando o interesse na classificação pela mesma e mineração de dados. Por isso, métodos que possam ajudar na classificação, recuperação, filtragem e controle desses dados estão sendo bastante explorados.

A classificação textual pode ser aplicada sob vários contextos e parâmetros sendo eles: a indexação dos documentos com vocabulário controlado, filtragem de documentos, geração automática de meta dados, desambiguação semântica, catálogos hierárquicos de recursos *Web* e qualquer tipo de aplicação que necessite da organização documental ou a seleção e adaptação de documentos (RIZZI, 2000).

As aplicações de categorização textual surgiram em meados dos anos 1960 expandindo-se para outras áreas como a recuperação e filtragem da informação, tornando-se uma área de pesquisa própria na década de 1980. Na década de 1990, devido aos avanços tecnológicos, surgiram *hardwares* mais potentes capazes de processar um grande volume de informações que combinadas com as técnicas de aprendizado de máquina começaram a popularizar-se e fazer parte do processo de categorização. Esses algoritmos de aprendizado de máquina passaram a apresentar resultados tão eficientes quanto à classificação realizada por especialistas.

Um problema enfrentado na classificação da informação ocorre devido ao grande volume de dados dispostos de forma desordenada e representados de várias formas, gerando uma grande quantidade de atributos para classificar um texto o que torna o processo de classificação proibitivo para a maioria dos algoritmos de classificação. Por isso, há a necessidade de filtrar esses atributos para não comprometer a qualidade da classificação, por isso, essa fase é de extrema importância no processo de classificação textual.

Neste trabalho tem como foco o levantamento de técnicas e ferramentas para o agrupamento de informações controladas, para agrupar as informações presentes em um produto em categorias, previamente cadastradas, com a finalidade de facilitar o processo de recuperação da informação em um *Marketplace*. Com isso desenvolver uma ferramenta que automatize o processo de categorização dos produtos, processando as informações textuais presentes em produto, agilizando o processo de categorização no *Marketplace*.

1.1.Motivação

Esse trabalho foi proposto dado ao grande volume de informações obtidas na indexação de variados tipos de produtos de diversas lojas virtuais que atuam em nichos de mercados diferentes e que estão em uma plataforma única de *marketplace*, tornando o processo de categorização de produtos custoso e ineficiente, dificultando ainda, o processo de organização dos produtos em suas devidas categorias, que por fim reduzem a navegabilidade no *Marketplace,* tornando difícil o processo de compra de produtos, e afetando a exibição dos produtos na plataforma.

Os problemas comuns identificados no processo de categorização são: grande parte dos usuários do sistema não obedece às regras de categorização do tipo "de-para" dentro de suas lojas virtuais; não há um padrão em nomear, hierarquizar uma categoria; há casos de lojas que trabalham com um público bem específico e, no *Marketplace* não existe uma categoria especializada, conforme a necessidade de tal cliente, mas há a possibilidade de classificá-la em uma categoria mais generalizada.

1.2. Objetivo

O objetivo desse trabalho é entender as diversas técnicas de agrupamento de dados, enfatizando as técnicas de classificação da informação, agrupando-as através de características conhecidas, suportando somente o agrupamento controlado de informações.

Realizar um estudo detalhado sobre as técnicas para a categorização textual levantando as informações necessárias para a execução da estruturação dos textos, com a finalidade de realizar o processamento desses documentos.

Estudar as diversas tecnologias que viabilizem o processamento de informações de forma ágil, e simples. Abstraindo técnicas as de agrupamento de dados e processamento textual.

Desenvolver uma ferramenta que realize o processamento das informações contidas nas características descritivas de um produto, sugerindo uma categoria que possa representar esse produto.

2. Categorização Textual

Categorizar pode ser definido como uma forma de agrupar objetos, substâncias ou qualquer entidade em grupos maiores que possuam uma temática conhecida e essa represente o objeto com uma maior abrangência.

Já a categorização textual nada mais é do que o agrupamento de textos com características semelhantes dentro de classe que possui um padrão pré-definido. É uma forma utilizada para organizar os documentos, com a finalidade de facilitar a recuperação da informação, reduzindo o tempo de busca de um documento, já que os mesmos estão contidos dentro de uma ou várias categorias específicas.

Rizzi et al. (2000) descreve que "a categorização de textos é uma técnica utilizada para classificar um conjunto de documentos em uma ou mais categorias existentes. Ela é, geralmente, utilizada para classificar mensagens, notícias, resumos e publicações. A categorização também pode ser utilizada para organizar e filtrar informações. Essa capacidade faz com que esta técnica possa ser aplicada em empresas, contribuindo no processo de coleta, análise e distribuição de informações e, consequentemente, na gestão e na estratégia competitiva de uma empresa".

A categorização textual surgiu em meados do ano de 1960, com o advento do uso dos computadores, mas passou a ter importância somente da década de 1990 com a expansão da internet que gerou um grande volume de informações combinadas com a capacidade de processamento dos computadores já existentes no período, ainda que, concentrado para algumas organizações.

O processo de categorização de textos compreende duas etapas: a definição das categorias e a categorização de documentos.

Na primeira fase, a definição das categorias é dividida em três etapas, sendo elas: preparação dos textos, seleção das características e a definição das categorias.

- A preparação dos textos é um processo de limpeza do texto, onde serão removidas palavras irrelevantes na classificação do texto, como stopwords, remoção de símbolos e outros, além do tratamento de inflexão de palavras.
- No processo de seleção de características são buscadas palavras ou conceitos que definam melhor as características dessas categorias utilizando métodos matemáticos para gerarem essas informações.

- Com isso é gerado um índice com as palavras ou termos comuns a todos os documentos. Esse índice representará, então, uma categoria.

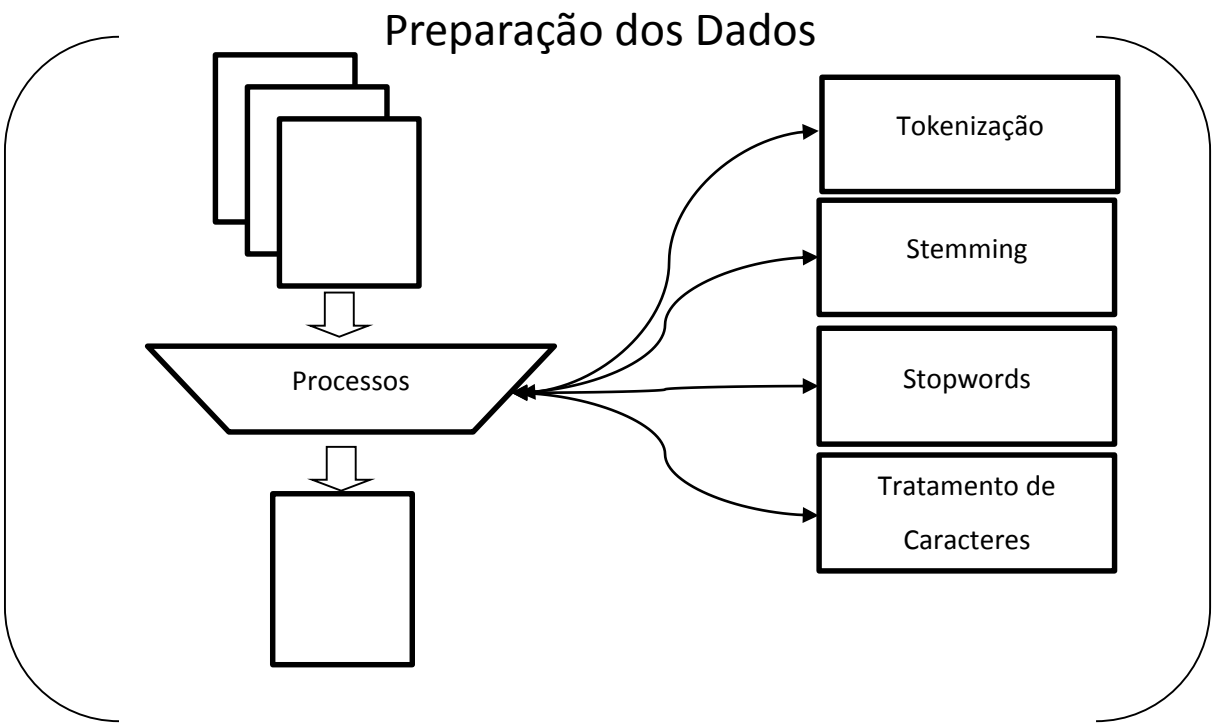
Já na segunda fase um novo texto a ser classificado passa pelos mesmos processos submetidos no texto-base anterior, submetendo a um processo de comparação entre os dois índices gerados e verificando a semelhança entre eles e assim determinando em qual categoria o texto deverá ser atribuído.

2.1 Processos executados na preparação do texto para a categorização

Estudos levantados comprovam que a fase do tratamento inicial do texto, claramente afeta o processo de categorização textual, é onde na maioria das situações pode provocar erros nos processos de categorização (GALHO, 2003).

É nessa fase que os dados são corretamente tratados removendo dados inconsistentes ou irrelevantes que possam prejudicar o processo de geração de informação, outra característica desta fase é a estruturação da informação, viabilizando o uso das técnicas de categorização textual. Na Figura 1 será ilustrado esse processo:

Figura 1 Esquema de preparação do texto



Fonte: Figura Adaptada (MEDEIROS, 2004)

2.2 Processamento dos textos

Consiste na fase de tratamento da informação, removendo palavras que não fornecem informações relevantes ao texto, união de palavras que possuam variações que apresentem o mesmo significado semântico entre outros tratamentos.

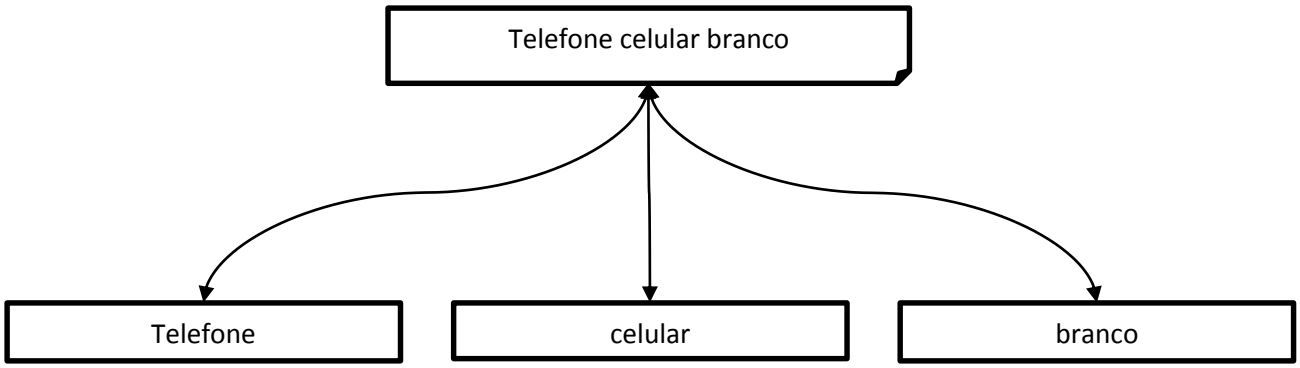
2.2.1 Tokenização

A tokenização nada mais é que, o processo de quebra do texto em várias partes, representadas por palavras, essas palavras são denominadas *tokens*.

O processo de quebra do texto é realizado utilizando caracteres delimitadores presentes em um texto tais como: “!”, “?”, “ ”, “(“, “)” entre outros.

A figura 2 demonstra o processo de funcionamento da tokenização:

Figura 2 Esquema representando a tokenização



Fonte: O próprio autor

2.2.2 Tratamento de stopwords

Na preparação do texto é necessária a remoção de palavras irrelevantes ao texto, como palavras denominadas *stopwords*. Estas são palavras que não trazem uma relevância significativa ao texto e ou representação para uma categoria. Em geral, são palavras auxiliares ou conectivas conhecidas como conjunções: e, ou, para, no, na, entre outras, que geralmente não fornecem nenhuma diferença significativa ao texto. Pode-se também considerar pronomes,

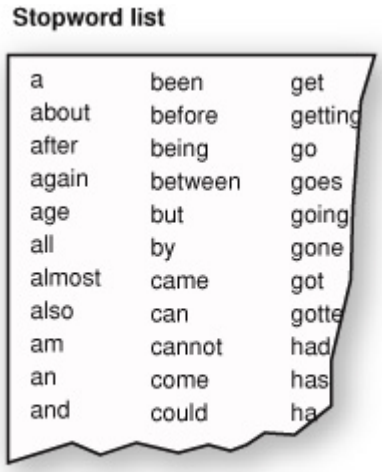
preposições que são palavras que apresentam alta incidência em um documento, mas, não afetam a categorização de um texto (MEDEIROS, 2004).

Para a execução será gerada uma lista dessas palavras, sendo geradas manualmente ou automaticamente, através de técnicas que verificam a recorrência das mesmas em um texto.

Esta técnica tem um papel fundamental no processo de categorização dos textos e deve ser executada meticulosamente, já que nele pode-se aumentar a velocidade do processo caso as remoções destas palavras sejam executadas corretamente ou, em outros casos, pode-se arruinar o processo de geração das categorias, removendo palavras relevantes ao texto.

A figura 3 representa uma lista de *stopwords*:

Figura 3 Representação de uma lista de *stopwords*



Fonte: Website Apple, 2014

2.2.3 Tratamento de caracteres especiais e ou inválidos

Nessa fase ocorre a eliminação dos caracteres inválidos ou irrelevantes ao processo de categorização com a finalidade de reduzir o tamanho do texto. Este processo ocorre através da remoção de caracteres inválidos ou especiais, a substituição de palavras acentuadas e também, a conversão do texto para caixa baixa.

2.2.4 Remoção de Prefixos e Sufixos (Stemming)

Essa técnica apresenta a característica de uma filtragem mais semântica do texto, ou seja, consiste em remover as variantes morfológicas de uma palavra, com a finalidade de melhorar a qualidade das palavras indexadas de um texto (GALHO, 2003).

Nesse processo há a indexação somente do radical que representa a palavra, removendo sufixos e prefixos de uma palavra, considerando regras gramaticais da linguagem utilizada. Existem alguns casos, em que a variação da palavra não é representada por um radical e para a representação dessas palavras é aconselhável o uso de uma lista de palavras que representa cada um desses radicais.

2.3 Representação do documento

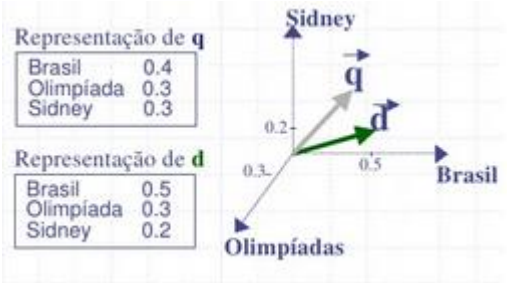
Após o processamento do documento as palavras geradas deverão ser organizadas, com a finalidade de representar o documento de forma compreensível ao processo de categorização textual.

2.3.1 Modelo de representação vetorial

O modelo de representação do documento utilizado é o vetorial, onde o documento é representando em um plano multidimensional. Atribuindo pesos para cada incidência de uma palavra ao documento representado em um plano. Nessas condições pode-se afirmar que textos com características muito parecidas possuem um alto grau de proximidade entre os vetores (MEDEIROS, 2004).

Na figura 4 é ilustrado um modelo de representação vetorial

Figura 4 Modelo de representação vetorial



Fonte: O próprio autor

2.3.2 Atribuição de pesos

Para Medeiros, no modelo apresentado na seção anterior vimos que os documentos são representados como vetores de termos e uma coleção de documentos é representando por uma matriz com uma dimensão igual ao número de documentos onde, cada termo, é representando por um elemento da matriz e um peso (MEDEIROS, 2004).

Considerando que os elementos da matriz estão relacionados com um peso que representa a frequência dele em um documento, há métodos que normalizam esses valores, distribuindo os pesos de forma mais representativas e, em relação a todos os documentos analisados, tratando alguns problemas comuns, como; a alta incidência de um termo em um documento e a sua relevância para o próprio termo ou mesmo quanto maior a sua incidência para uma coleção de documentos menos importante será para a coleção de documentos.

2.3.2.1 Booleano

Atribui-se pesos a um termo do documento, utilizando os valores zero ou um para representar a ausência ou presença de um termo no documento. Conforme está representado na tabela 1.

Tabela 1 Classificação booleana

	colorido	**texto**	**branca**
caneta	1	1	0
lápis	1	0	0
borracha	0	1	1

Fonte: o próprio autor

2.3.2.2 TF (*Term Frequency* - tf)

Segundo Manning *et al.* (2008) neste método é considera a quantidade de ocorrência dos termos no texto, onde palavras com maior incidência possuem uma maior relevância ao documento, o que é um argumento inverídico, já que um termo pode ocorrer dez vezes em um documento e em outro termo somente uma vez. Isso não indica que o primeiro termo possui uma maior relevância para o resultado.

2.3.2.3 IDF (*Inverse Document Frequency* - idf)

Esse método baseia-se na contagem de documentos na coleção buscada ou indexada, contendo o termo em questão. Intuitivamente, o termo buscado ou indexado, em muitos documentos, possui uma menor relevância para o restante da coleção, não sendo um bom discriminador e, por isso, esse termo deve possuir um menor peso em relação a um termo que possui uma menor recorrência e são esses termos que melhor representam esses documentos (ROBERTSON, 2004).

$$idf_t = \log \frac{N}{\text{tf}}$$

Onde:

N: número de documentos na coleção;

tf: *Term Frequency* ;

t: termo;

2.3.2.4 TFIDF (*Term Frequency/ Inverse Document Frequency*)

O modelo TFIDF considera a frequência de cada termo em todos os documentos, como medida inversa da sua capacidade de representar especificamente cada documento, ou seja, em quanto mais documentos o termo aparecer menos representativo na especificação do documento o mesmo será. Se um termo aparece cinco vezes mais em um documento do que em outro, não se pode concluir nada a respeito de sua relevância, pois o tamanho dos textos pode contribuir para a ocorrência maior em um deles. Todavia, se um termo aparecer cinco vezes mais em documentos da coleção que outro termo aparece, esse termo que aparece em menos documentos terá um maior peso de decisão na categorização (MEDEIROS, 2004).

2.4 Métodos de categorização

Este tipo de aprendizado consiste em analisar um conjunto de situações, denominadas “instâncias” ou exemplos e, suas características, denominadas “atributos”. Entre os atributos, aquele que apresenta a solução previamente conhecida é denominado “atributo alvo” ou “rótulo de classe”. Os valores do atributo alvo constituem as soluções previamente conhecidas e, portanto, compõem o conjunto de possíveis soluções (FURQUIM, 2011).

2.4.1 Espaço vetorial e similaridade de cosseno

O modelo representado considera os valores indexados como vetores em um espaço multidimensional, onde as similaridades de dois textos são mensuradas através do cosseno entre eles. Quanto menor o ângulo entre eles, maior é a similaridade.

2.4.2 Redes Neurais

Utiliza-se a técnica de redes neurais que têm como característica identificar padrões entre os textos indexados. Cada neurônio possui um peso de ativação para o treinamento da rede e que são definidos constantes de treinamento e a taxa de aprendizado do neurônio. Então, a cada inserção de textos, os pesos de ativação são recalculados até que ocorra a estabilização dos pesos de ativação, onde caracteriza a rede neural como treinada. Após o treinamento, a rede está pronta para a categorização de novos textos (GALHO, 2003).

2.4.3 Similaridade Difusa

A lógica difusa, proposta por Zadeh (1967), na categorização de textos, propõe a solução para o problema de ambiguidade, porque trata das situações imprecisas, melhorando o resultado através do cálculo de pertinência de um conjunto. Com o uso dessa técnica pode-se determinar o quanto um termo é importante para a categoria utilizando operadores booleanos ampliados para o tratamento de incertezas.

O conjunto de tuplas representando o texto normalizado, com o peso do termo e um valor difuso definido entre zero e um e indica a importância do termo, ou seja, quanto mais próximo o valor de um mais relevante será o termo.

Assim baseando-se na ideia de similaridade, permite-se que os resultados ofereçam não apenas, classificações exatas de um documento, mas também, gerem relações parciais da classe. Atribuindo um grau de pertinência ou relevância de uma classe em relação a um texto tratado.

2.4.4 KNN – Vizinho mais próximo

O método do vizinho mais próximo realiza a classificação, baseado nos vizinhos mais próximos. Partindo do princípio de que o texto não precisa ser necessariamente igual, mas, possuir características mais próximas. É necessária uma base de exemplos boa para o treinamento, de forma que não ocorram distorções no processo de classificação. Mostrou-se eficiente na classificação de produtos com uma base de dados grandes (GALHO, 2003).

2.4.5 Árvores de decisão

A classificação ocorre através das regras de decisão, as árvores são compostas por nodos e ramos, cada nodo é uma classe, exceto os terminais, representando uma regra de decisão e os ramos em possíveis sub árvores, estes representam cada possível resultado deste teste de decisão. Os nodos terminais representam a classe dessa decisão. Ela pode ser utilizada quando existem dependências entre as categorias e onde as categorias mais especializadas estão mais próximas dos nodos terminais (GALHO, 2003).

2.4.6 Modelos de regressão

No modelo de regressão, as categorias são treinadas através de um conjunto de textos previamente categorizados, representados em duas matrizes: a primeira é uma matriz de textos e seus elementos são os pesos; a segunda é uma matriz de categorias e os seus elementos são os pesos das categorias (GALHO, 2003).

2.4.7 Naives Bayes

Ele utiliza a probabilidade com que uma palavra pode aparecer em um documento, sabendo previamente a categoria que ele pertence, ou seja, pode-se definir uma probabilidade de um documento pertencer a uma categoria dentro das estruturas hierárquicas. E também definir a probabilidade das diferentes categorias serem atribuídas a um documento; a de maior probabilidade é a escolha para ser atribuída (GALHO, 2003).

2.5 Considerações Finais

Nesse capitulo, abordou-se o que é categorização e também foi abordada a categorização textual; tema principal utilizado no processo de categorização de produtos.

A complexidade observada no levantamento das informações do processo de categorização textual fará com que a abordagem do assunto no desenvolvimento do capítulo seja mais simplificada, para facilitar o processo de entendimento do leitor.

Visualizou-se que nas diversas etapas do processo de categorização textual, a fase inicial de limpeza do texto será uma fase imutável, ou seja, possui poucas variações não sendo citadas outras técnicas para este processo.

Neste capítulo também fora possível definir e entender algumas técnicas que poderiam ser utilizados no processo de desenvolvimento do sistema de categorização textual, como atribuição de pesos ou seleção de características observou-se, ainda, uma maior eficiência da técnica de TFIDF, já que a mesma considera uma maior quantidade de parâmetros para atribuir relevância ao termo em um documento na representação de uma categoria. Quanto aos métodos de categorização levantados fora observado uma alta complexidade nesses algoritmos, verificando a necessidade do uso de uma ferramenta que auxilia no processo de categorização textual.

No próximo capítulo serão levantadas algumas ferramentas utilizadas no processo de categorização.

3. Tecnologias para categorização

Neste capitulo serão descritas as tecnologias levantadas que poderão ser utilizadas no processo de categorização textual, auxiliando no processo de categorização, bem como, no processo de redução e contagem de termos, distribuição e controle no processamento dos dados em máquinas distribuídas e na execução do processo de categorização, utilizando bibliotecas que já implementem essas técnicas.

3.1 Map Reduce

É um modelo proposto para processar uma grande capacidade de informações paralelamente. Seu objetivo é facilitar a programação de aplicativos distribuídos (DEAN GHEMAWAT, 2004).

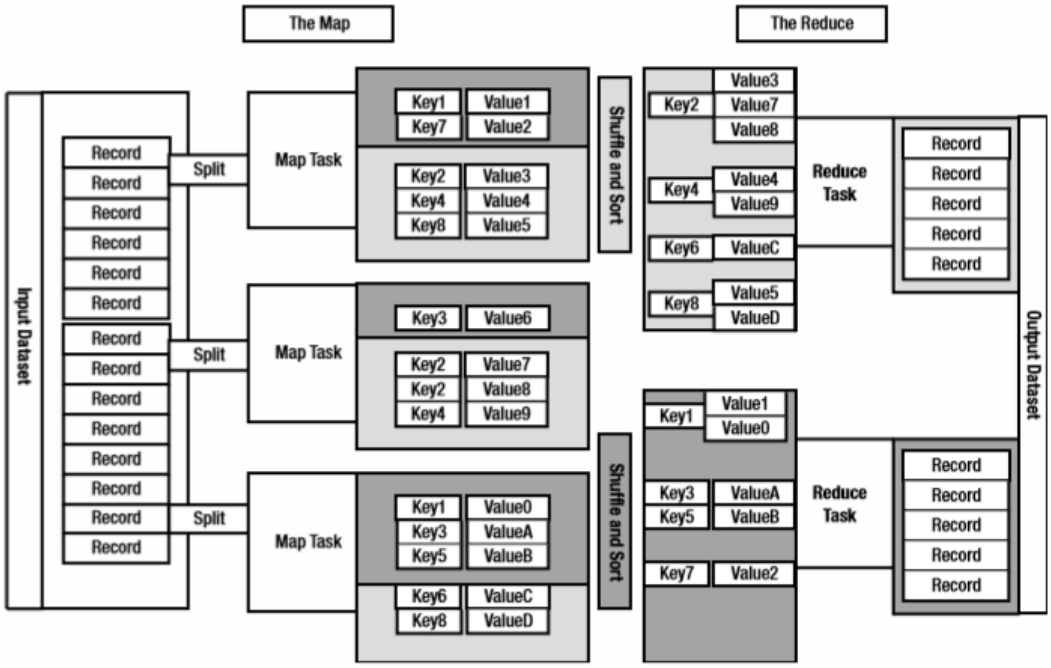
Modelo criado pela Google baseia-se no conceito de: interação ao invés de entradas; registro de chave/valores para parte da entrada; agrupamento de informações por chave; interação sobre os resultados agrupados; redução de cada grupo (DEAN GHEMAWAT,2004).

Esta abordagem foi adotada, pois, verificou-se que em muitos casos, era necessário mapear fragmentos de dados de entrada por uma chave identificadora e, então, compartilhar todos os fragmentos que usassem a mesma chave. Assim, a tarefa principal do programador é implementar estas duas funções; indicando como o mapeamento e a redução dos dados serão compostos.

Como a representação das informações é simples e efetiva dando suporte ao paralelismo no processamento da informação, isso dá ao desenvolvedor uma visão mais abstrata do problema da computação paralela e distribuída porque a própria implementação do *MapReduce* é responsável pelo balanço de carga, performance de rede e tolerância à falhas.

O principal papel do *MapReduce* é descrever um modelo baseado em vários clusters de máquinas *comodities* com dados locais. Esse modelo de programação pode parecer um restritivo, porém, encaixa-se em muitos problemas encontrados na prática de processar uma grande quantidade de dados. Também possui uma grande limitação que é suprida através da decomposição do problema em múltiplos processos de *MapReduce*.

Figura 5 Funcionamento do *Map Reduce*



Fonte: Pro Hadoop

O fluxo do *Map Reduce* pode ser descrito da seguinte forma; Entrada de dados:

Os dados são divididos em tamanhos iguais dependendo da configuração utilizada ou hardware utilizado. Cada pedaço é, então, passado para uma função Map.

- Função *Map*: A função *map* recebe uma parte da informação e processa-a gerando uma série de dados do tipo chave-valor, como no caso de registro de visitas de uma pessoa a um produto ele informa a quantidade de vezes que uma determinada pessoa visualizou uma informação.
- Função de particionamento: Cada saída da função *map* é alocada para uma função de redução. Isso dá ao sistema uma distribuição uniforme dos dados por *shard.*
- Função de comparação: Os dados emitidos pela função *map* são organizadas através de uma função de comparação.
- Função *Reduce*: O framework chama a função *reduce* para cada chave única ordenada. A função *reduce* percorre os valores e os associa aos outros, produzindo zero ou mais saídas.

- Saída para escrita: Os dados de saída da função *map* são escritos em um sistema de arquivo distribuído.

3.1.1 Hadoop

É um projeto *open-source* da Apache Software Foundation, que possibilita o processamento de grande quantidade de dados, sobre clusters de servidores commodities (IBM DEVELOPER WORK, 2014).

Foi desenvolvido para funcionar em um único servidor ou milhares de máquinas com um alto grau de tolerância a falhas. O seu alto grau de tolerância à falhas dá-se devido a sua habilidade em detectar os erros na camada de aplicação.

Os módulos presentes no Hadoop são: Hadoop Distributed File System (HDFS) *e Map Reduce.*

Hadoop Distributed File System (HDFS)

É o sistema de arquivos que pode ser armazenado em grandes *datasets*. Ele é escalado em todos os nós locais que fazem dele um grande sistema único de arquivos. Uma das características desse sistema de arquivos estará em sua disponibilidade em vez da redundância das informações.

A arquitetura do modelo HDFS é o mestre/escravo. Onde há somente um sistema no mestre que possui as seguintes responsabilidades:

- Abrir, fechar e renomear arquivos e pastas;
- Mapear os blocos para os nós escravos;
- Gerenciar espaços;
- Mapear todos os escravos para a distribuição das tarefas;
- Os nós escravos possuem as seguintes responsabilidades:
- Operações de criar, remover e replicar os blocos de informações;
- Leitura e escrita do sistema de arquivos.

Hadoop MapReduce

É a implementação do modelo de programação descrito na seção de *Map Reduce*.

3.2 Apache Lucene

O Apache Lucene é uma ferramenta *open-source* desenvolvido pela Apache Software Foundation. Na sua definição ela é uma biblioteca para a indexação e recuperação de informações em documentos, possui um alto desempenho, fácil configuração e é altamente portável, tornou-se bastante popular tanto no campo comercial, quanto no campo acadêmico (LUCENE 2014).

A arquitetura lógica utilizada no Apache Lucene é baseada no conceito de separação do texto em *tokens*, atribuindo pesos conforme a frequência de ocorrência do termo nos textos, caracterizando o Apache Lucene como uma ferramenta de busca textual (SANTOS, 2012).

Algoritmos suportados na busca textual:

- *ranked searching -- best results returned first*
- *many powerful query types: phrase queries, wildcard queries, proximity queries, range queries and more*
- *fielded searching (e.g. title, author, contents)*
- *sorting by any field*
- *multiple-index searching with merged results*
- *allows simultaneous update and searching*
- *flexible faceting, highlighting, joins and result grouping*
- *fast, memory-efficient and typo-tolerant suggesters*
- *pluggable ranking models, including the Vector Space Model and Okapi BM25*
- *configurable storage engine (codecs)*

3.3 Mahout

O Apache Mahout é um projeto *open-source* da Apache Software Foundation, que possui várias bibliotecas de aprendizado máquina, utilizando máquinas escaláveis (THE APACHE FOUNDATION,2014).

Foi um projeto derivado do Apache Lucene, que é um projeto da Apache Software Foundation e, que implementa técnicas avançadas de busca, mineração de textos e recuperação

da informação. Como estas técnicas são muito similares, houve uma separação dos projetos para dar início ao projeto Apache Mahout, absorvendo com o tempo o projeto do Apache Taste (OWEN et al, 2011).

Projeto *Mahout* é considerado um *framework* escalável, já que, utiliza as informações processadas via *Hadoop* utilizando o *Map Reduce*, armazenadas pelo sistema de arquivos HDFS que estão distribuídos entre os clusters e nativos.

Ele dá suporte a três tipos de casos :

- Filtragem Colaborativa (Recomendação) faz comparações entre o comportamento entre os usuários, fornece outros resultados que possam interessar a outros usuários. Utiliza as seguintes técnicas ou algoritmos para recomendar os produtos: *User-Based Collaborative Filtering; Item-Based Collaborative Filtering; Matrix Factorization with Alternating Least Squares, Matrix Factorization with Alternating Least Squares on Implicit Feedback; Weighted Matrix Factorization, SVD++, Parallel SGD*.
- Agrupamento (*Clustering*) que nada mais é que uma técnica de agrupamento das informações, através somente da entrada de dados, sem estabelecer um controle da forma que as informações devem ser agrupadas. Utiliza as técnicas de agrupamento: *Canopy Clustering; k-Means Clustering; Fuzzy k-Means; Streaming k-Means; Spectral Clustering*.
- Categorização que agrupa as informações de forma controlada, através de documentos já categorizados, buscando agrupar novos produtos dentro dessas categorias inseridas e já treinadas. Dá suporte às seguintes técnicas: *Logistic Regression; Naive Bayes; Random Forest; Hidden Markov Models; Multilayer Perceptron*.

3.4 Weka

O Weka é uma coleção de algoritmos de *machine learning* para as tarefas de mineração. Os algoritmos podem ser aplicados direto nos *datasets* ou chamadas através de uma API Java. Possui ferramentas para pré-processamento dos dados, classificação, regressão, *clustering* ou regras de associação e também a visualização da informação. É adequada para o desenvolvimento de novos *schemas* de aprendizado-máquina (WEKA, 2014).

A principal desvantagem encontrada na ferramenta Weka está ligada ao processamento de dados utilizando máquinas escaláveis, apesar de possuir algumas ferramentas que utilizam as técnicas de *Map Reduce* distribuídos e existe a possibilidade de integração com o Hadoop para utilização de máquinas *comodities*. Não possui um desenvolvimento tão integrado como o Apache Mahout.

3.5 Considerações Finais

Com o levantamento das tecnologias observadas nesse capitulo, será possível determinar quais ferramentas e técnicas deverão ser utilizadas para o processo de categorização textual.

Todas as ferramentas serão utilizadas e testadas como objetivo de conhecê-las as e determinar a possibilidade de uso das mesmas em sistemas de produção, ou seja, verificando a possibilidade do uso das ferramentas em ambiente real. Por isso, foi considerada a comunidade envolvida no desenvolvimento das ferramentas, o suporte disponível às ferramentas, a quantidade de usuários que já utilizam as ferramentas e a facilidade de integração entre elas quando necessário.

A ferramenta Weka surgiu como uma opção para a substituição ao Mahout, mas realizar integração da mesma com o Hadoop dependia de outras bibliotecas, já que essa ferramenta não é uma implementação nativa para o uso em computação distribuída, outro fator que influenciou no processo decisório fora a quantidade de usuários e suporte encontrado na *internet* em favor do uso do Apache Mahout.

Por isso, no processo de implementação do categorizador, serão utilizadas as ferramentas Apache Hadoop, Apache Mahout e Apache Lucene, onde o Hadoop será a ferramenta responsável em realizar o processamento distribuído e execução do *Map Reduce*. O Apache Lucene será o responsável na realização do processamento do stemmer, através dos analisadores, presentes no texto com suporte à Língua Portuguesa e o Mahout a ferramenta responsável pela categorização.

No próximo capítulo será demonstrado como ocorre o processo de integração entre essas ferramentas e o funcionamento do sistema categorizador.

4. Metodologia

Nesta capitulo são descritas as etapas desenvolvidas da ferramenta para o processamento do conteúdo textual do categorizador de produtos.

4.1 Categorização Automática de Produtos

Este trabalho buscará soluções para um problema encontrado em um *marketplace* para classificar os produtos oriundos de várias lojas virtuais integradas ao sistema e que não respeitam as regras de categorização de produtos utilizadas no mesmo ou possuem uma nomenclatura errada desses produtos, ou ainda, seguem um padrão de nomenclatura totalmente diferenciada.

Foram realizadas tentativas de categorizar os produtos, inicialmente será desenvolvido um categorizador automático de produtos, que realizará uma busca no banco de dados pelo título do produto sem tratamento nenhum dos termos e assim, os agrupará em categorias mais relevantes que possuísse o título do produto. Deste modo, será realizado um cálculo para atribuir ou não um produto a uma categoria ,entretanto, essa ferramenta apresentará um baixo índice de assertividade que ocasionará uma alta taxa de correções e a necessidade de uma manutenção constante dos produtos, por conta dos erros que ocorrem na categorização de um produto e ,portanto, diminuiria ainda mais o índice de assertividade da ferramenta.

Atualmente, não existe nenhuma regra ou solução agrupar os produtos integrados ao *marketplace*, os produtos recebem uma categoria genérica e são exibidos sem qualquer critério de classificação e a fim de atender a demanda de alta quantidade de produtos, ainda que não classificados.

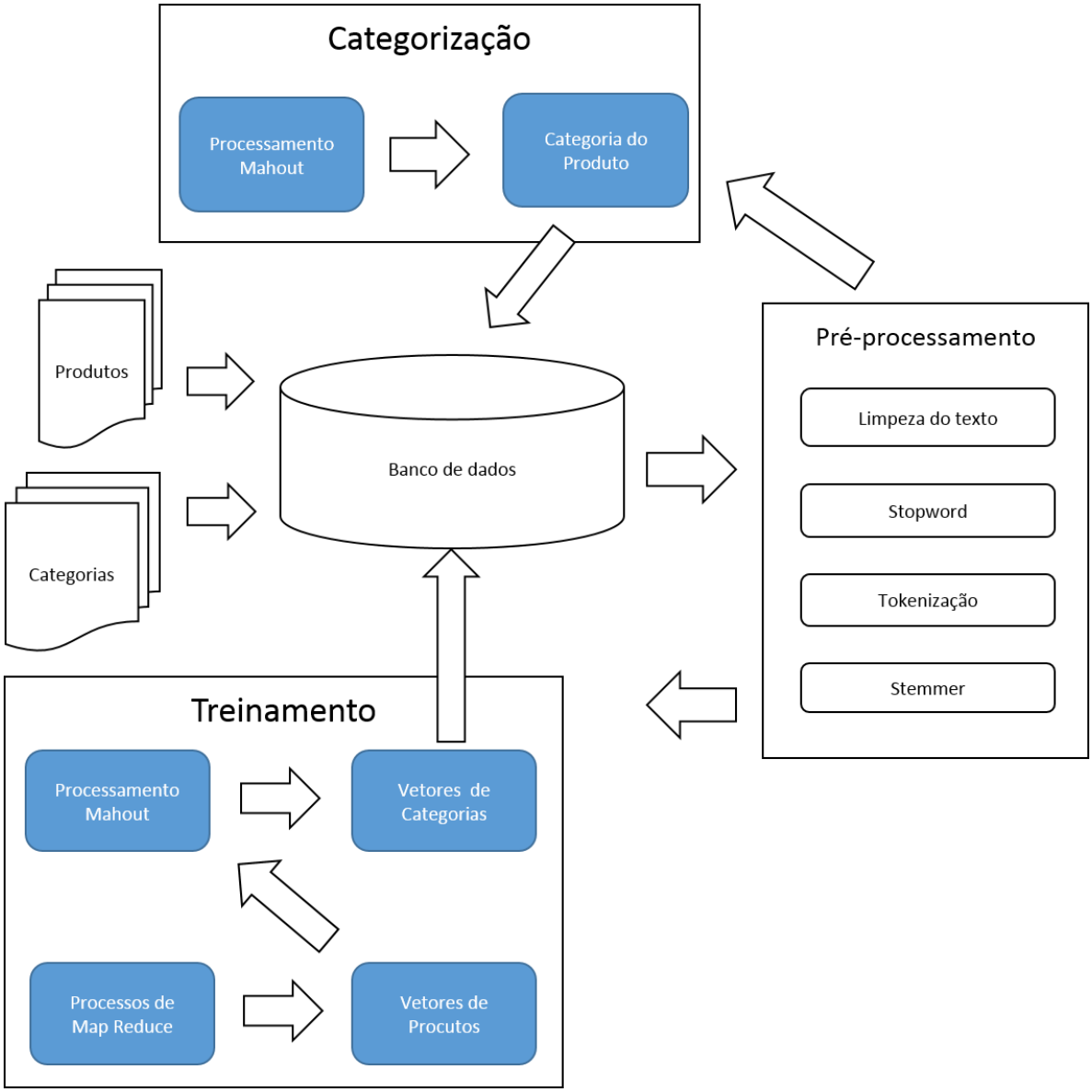
Como há a necessidade de categorizar os produtos para facilitar o processo de recuperação destes, haverá a contratação de um individuo X que vai ser responsável pela categorização dos produtos manualmente, porém, tal processo demonstrar-se-á pouco eficiente, dada a enorme quantidade de produtos existentes e o alto giro de produtos indexados diariamente no *marketplace*.

As soluções utilizadas para resolver a categorização de produtos como se observa serão pouco eficientes, e por isso neste trabalho fora proposto o desenvolvimento de um novo categorizador de produtos, que utilizaria regras de categorização textual, com o objetivo de solucionar o problema na recuperação de produtos.

4.2 Ambiente

Para a solução do problema de categorização de produtos foi proposto um esquema de interações entre as diversas etapas presentes no processo de categorização. Com a finalidade de abstrair ao máximo as interações ocorridas entre o Apache Hadoop e o sistema de arquivos utilizado por ele, evidenciando somente o uso das ferramentas desenvolvidas para o categorizador, como todas as iterações o Map Reduce, as trocas de informações do banco de dados e o processo de limpeza as informações, com a finalidade de facilitar o entendimento da aplicação. Esse ambiente é descrito na figura 06:

Figura 6 Esquema de funcionamento



Fonte: O próprio autor

4.3 Etapas do processo

O mecanismo de categorização foi divido em quatro etapas principais, a saber:

Etapa de cadastro de informação

O processo de cadastro de informação consiste na entrada de informações que serão utilizadas no processamento da mesma para a geração de um vetor de categorias na execução do processo de categorização de um produto.

Etapa – Pré-processamento (Limpeza)

Esta etapa consiste no processo de limpeza do texto, sendo assim, limpeza dos dados, removendo palavras que não oferecem relevância aos textos, radicalização de termos (*stemming*) e remoção de caracteres inválidos.

Etapa – Treinamento

Nesta fase ocorre o treinamento do categorizador, ou seja, são gerados diversos modelos representando as categorias dos produtos. A característica desta fase consiste no processo de cadastros de produtos que já foram categorizados manualmente com o objetivo de calibrar os modelos de categorias.

Etapa - Categorização

Nesta etapa será executado o processo de categorização, onde o vetor representando um produto é analisado e, então, será calculada a probabilidade entre os diversos modelos de representação de categorias, geradas na fase de treinamento na busca daquele modelo que representará uma maior probabilidade de representar a categoria e valor utilizado.

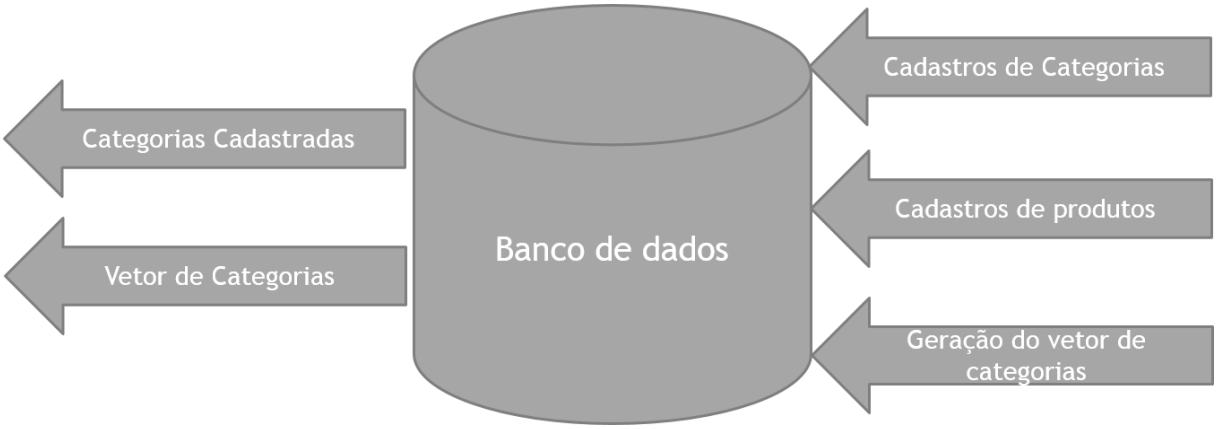
4.4 Cadastros de informação

O processo de cadastro de informação consiste na entrada de informações que serão utilizadas no processamento das mesmas para a geração de um vetor que representará as categorias na execução do processo de categorização de um produto.

O processo inicia-se com o cadastrado de uma categoria, contendo o nome e uma chave de identificação da categoria. Após esse processo são cadastrados diversos produtos que já possuem a categoria conhecida para que haja a geração do vetor de representação de categorias.

Na Figura 7 é ilustrado esse processo:

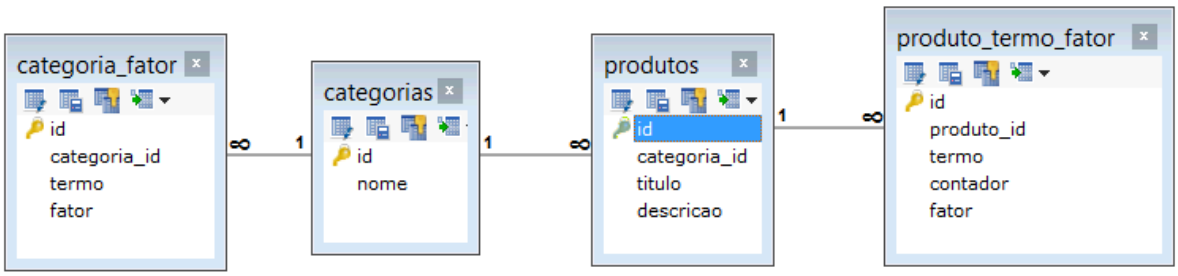
Figura 7 Esquema de Cadastro de informações no banco de dados



Fonte: O próprio autor

Esta etapa consiste em realizar o cadastro das categorias e produtos. Os dados salvos no processo de cadastro de categorias e produtos serão realizados conforme o esquema ilustrado na figura 8:

Figura 8: MER para representação de produtos e categorias



Fonte: O próprio autor

Na tabela de categorias serão cadastradas as que deverão ser treinadas. Na tabela produtos serão cadastrados os produtos que caso haja a necessidade de treinamento de uma categoria, esta deverá ter sido informada no campo de categoria_id.

As tabelas categoria-fator armazenam o vetor de representação de uma categoria e a produto termo-fator armazena os valores do map reduce para cada produto processado, tanto no processo de map reduce, quanto na atribuição de peso do produto.

4.5 Pré-processamento (limpeza)

Nesta etapa serão implementadas quatro classes responsáveis pela limpeza do texto. A classe *Tokenizer* é a responsável pela a tokenização das palavras, ou seja, é a classe responsável pela quebra de palavras e essa quebra será realizada considerando apenas a separação por espaços.

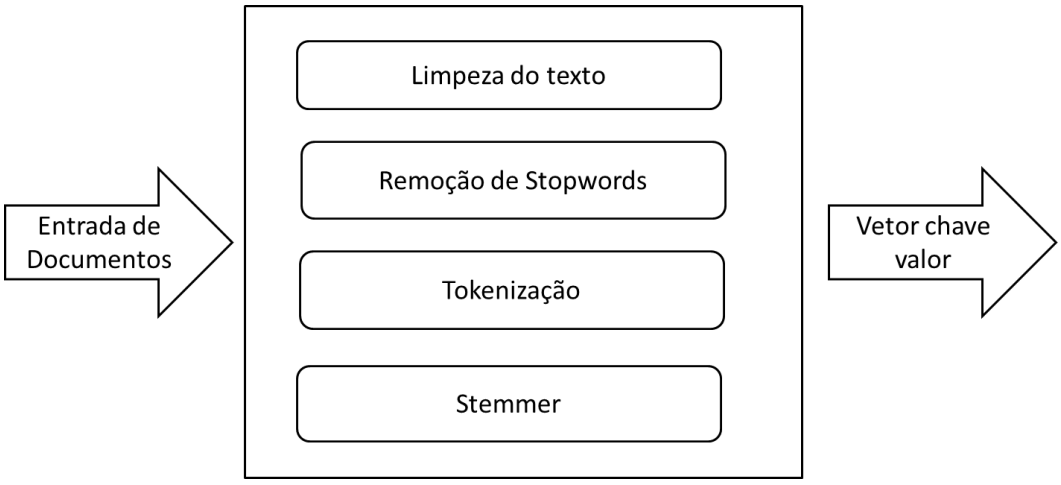
Na classe SpecialChar será implementada para a troca de caracteres especiais como acentos e cedilha e à remoção de caracteres de pontuação como: “;”, “,”, “:”, “?” entre outros.

Já a classe de *Stemmer* será responsável em radicalizar os termos, ou seja, executar a redução da palavra para o seu menor radical e será utilizada a biblioteca *PortugueseMinimalStemmer* presente no Apache Lucene para realizar esse tratamento de Stemmer.

E a última classe denominada *Stopwords* vai ser responsável pela remoção de stopwords a qual existe uma lista presente na Língua Portuguesa.

Na Figura 9 é ilustrado o diagrama de classe do Pré-Processamento.

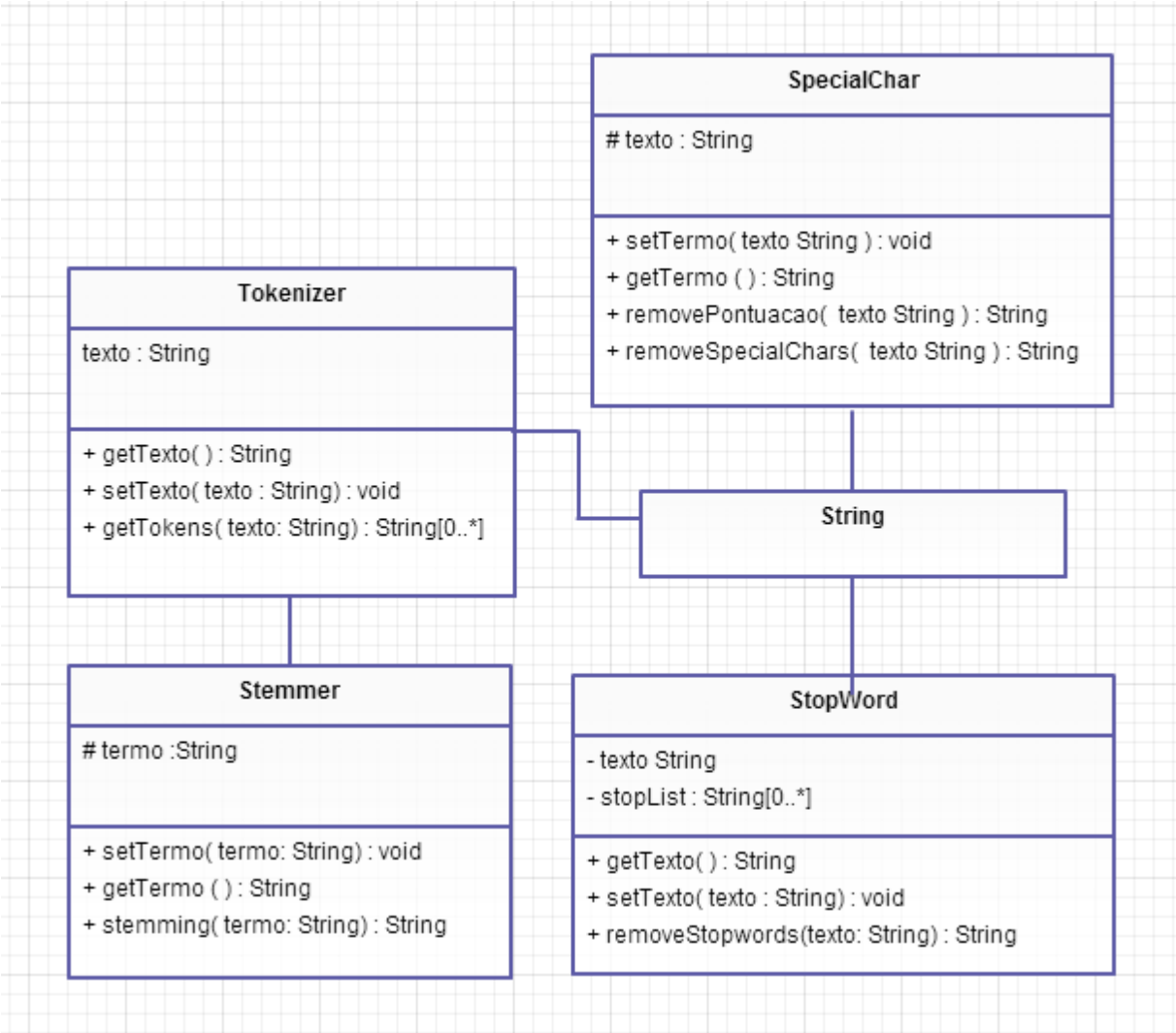
Figura 9 Diagrama de classe do Pré-Processamento



Fonte: O próprio autor

Na Figura 10 é ilustrado o diagrama de classe do Pré-Processamento.

Figura 10 Diagrama de classe do Pré-Processamento



Fonte: O próprio autor

4.6 Treinamento

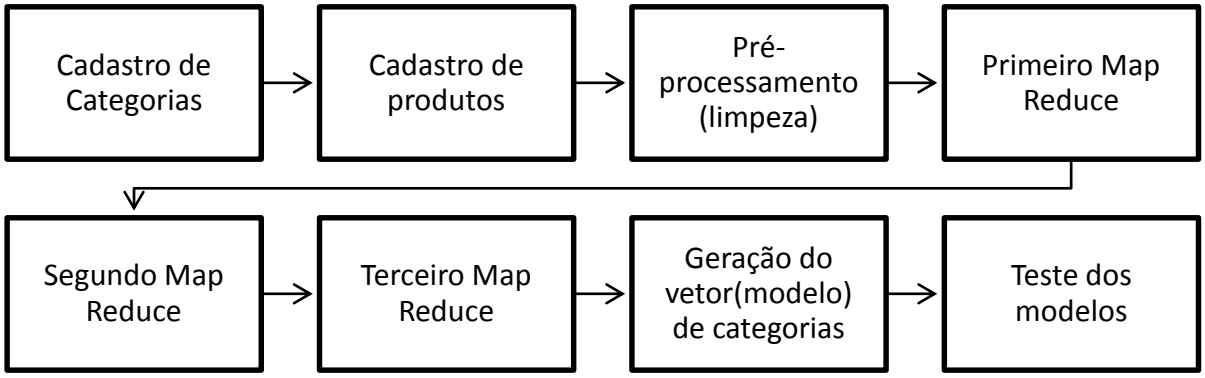
A etapa de treinamento compreende a execução do *Apache Hadoop* no processamento dos títulos e descrição do produto, conquanto a primeira execução do *Map Reduce* será somente para a quebra e contagem dos textos. Na segunda execução, haverá a totalização dos resultados e na terceira serão atribuídos os pesos no vetor de produtos, utilizando a técnica de TFIDF.

Após a fase de geração dos vetores será, então, utilizada a API do *Apache Mahout*, que executará o processo-treinamento. Utilizando a técnica de categorização de produtos Naive

Bayes processará as informações dos produtos tratadas no processo anterior, fazendo a associação probabilística entre os vetores dos produtos tratados e as categorias associadas e, ainda, utilizando técnicas de processamento e *Map Reduce*, com a finalidade de gerar um modelo de vetorial de cada uma das categorias dos produtos utilizadas. Este processo ocorrerá de forma implícita e nele serão executadas duas fases; na primeira fase será executado o treinamento às diversas categorias relacionadas a todos os produtos cadastrados e, no segundo, será executada uma fase de teste que utiliza uma menor quantidade de produtos, desta maneira, serão realizados testes de categorização dos produtos. Após a observação de que processo de categorização funcionou adequadamente, ou seja, o processo atingiu uma porcentagem satisfatória de categorizações corretas o vetor das categorias poderá ser armazenado no banco de dados para ser utilizado no processo de categorização dos mesmos.

Na figura 11 ilustra o fluxo do processo de treinamento.

Figura 11 Fluxo do processo de Treinamento



Fonte: O próprio autor

As fases do processo de treinamento são explicadas passo- a -passo abaixo:

- No início será realizado o cadastro das categorias que serão treinadas;
- Serão cadastrados produtos com as categorias já conhecidas;
- Será executado o processo de pré-processamento responsável pelo tratamento do texto;
- Ocorrerá à primeira iteração do *Map Reduce* através da ferramenta Apache Hadoop, que realizará a contagem dos termos;
- Após o processo de contagem ocorrerá outra execução do *Map Reduce*, só que neste momento será realizado totalização dos termos para o processo de atribuição de pesos;

- Na terceira iteração do processo de *Map Reduce* será então calculado o peso de cada termo referente ao produto na geração do vetor representando o produto;
- É gerado o modelo vetorial das categorias, através da ferramenta Apache Mahout e será salvo no banco de dados.
- São realizados testes, verificando se os modelos de categorizas conseguem classificar corretamente produtos com categorias conhecidas.

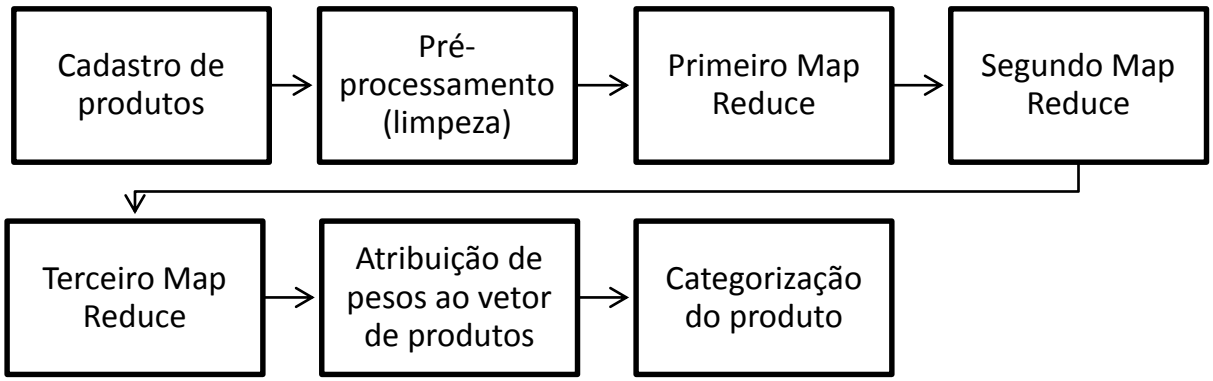
4.7 Categorização

Na execução de categorização serão feitas as mesmas funções de redução de um produto da fase de treinamento, ou seja, será executada, além do processo de limpeza dos textos dos produtos, também, a execução dos processos de redução com a finalidade de gerar o modelo vetorial que representará cada produto.

É nesta fase que ocorrerá o processo de categorização utilizando o modelo vetorial gerado na fase de treinamento das categorias. O modelo de vetorial será recuperado do banco de dados e as informações serão utilizadas na biblioteca Apache Mahout e, portanto, realizada a comparação dos diversos vetores e verificação de qual categoria representará melhor o produto a ser categorizado. Como acontece na fase anterior, esse processo também ocorrerá de forma implícita, somente retornando a categoria que melhor enquadra o produto.

A figura 12 ilustra o funcionamento do processo de categorização de produtos.

Figura 12 Fluxo do processo de categorização



Fonte: O próprio autor

As fases do processo de categorização são detalhadas abaixo:

- O início do processo dar-se-á através do cadastro de um produto sobre o qual se desconhece a categoria;
- Será executado o processo de pré-processamento responsável pelo tratamento do texto;
- Ocorrerá à primeira iteração do *Map Reduce* através da ferramenta Apache Hadoop, que realiza a contagem dos termos;
- Após o processo de contagem ocorrerá outra execução do *Map Reduce*, só que neste momento será realizada a totalização dos termos para o processo de atribuição de pesos;
- Na terceira iteração do processo de *Map Reduce* serão, então, calculados os pesos de cada termo referente ao produto na geração do vetor representando o produto;
- Utilizando a API do Apache Mahout os modelos vetoriais das categorias serão recuperados no banco de dados e, desta maneira, o produto será processado implicitamente, sendo possível somente visualizar a categoria com maior probabilidade de representar o produto.

4.8 Considerações Finais

Neste capítulo foram descritas as metodologias utilizadas na implementação do categorizador de produtos.

Devido à complexidade observada na fase de pesquisa sobre as técnicas de categorização textual, também fora observada no processo de desenvolvimento da ferramenta. Houve a necessidade filtrar as informações que eram realmente pertinentes ao processo de categorização, a fim de facilitar o entendimento do desenvolvimento da ferramenta.

O uso dos *frameworks* da Apache Foundation exigirão um estudo detalhado e minucioso, para melhor entendimento do funcionamento e como devem ser implementados e utilizados corretamente, bem como, as bibliotecas de cada uma das ferramentas. A maior dificuldade enfrentada será então no que tange a utilização do Apache Hadoop, exigindo um conhecimento técnico na configuração do *framework*, na utilização do seu sistema de arquivos distribuídos e na integração das informações oriundas do banco de dados sem o uso de outra ferramenta que realizasse essa integração.

No próximo capítulo, serão demonstrados os resultados obtidos nas diversas etapas do processamento do texto.

5. Resultado

De forma a validar a proposta será definida e utilizada uma base de dados com 2679 produtos, organizados nas categorias: Smartphones, Televisores e Bebidas.

5.1 Resultados da etapa de pré-processamento.

Para ilustrar um resultado da fase de pré-processamento serão escolhidos dois textos distintos, escolhidos aleatoriamente.

Cenário 1 contêm 164 palavras referentes ao tema bebida: Na figura xx está ilustrado o texto antes do pré-processamento.

A figura 13 demonstra o texto do cenário 1 sem tratamento

Figura 13: Cenário 1 sem tratamento

VINHO ESPANHOL FERNANDO DE CASTILLA CLASSIC DRY SEC FINO / PREMIUM CREAM / CLASSIC OLD MEDIUM AMONTILLADO 750ml. VINHO ESPANHOL FERNANDO DE CASTILLA CLASSIC DRY SEC FINO 750ml. Um clássico Jerez Fino, com aroma de frutas secas e o inconfundível toque pungente conferido pelas leveduras (flor), mesclado a um sutil toque de oxidação. Seco, com boa acidez e muito equilibrado, tem sabor marcante e longa persistência. Acompanha bem os "tapas" espanhóis e frios, além de amêndoas secas e levemente tostadas. VINHO ESPANHOL FERNANDO DE CASTILLA PREMIUM CREAM 750ml. Vinho delicadamente doce, com aroma de frutas secas, especiarias e carvalho tostado Macio e concentrado, tem longa persistência e sutil retro-olfato. Excelente com sobremesas e bolo de frutas secas (nozes e avelã). VINHO ESPANHOL FERNANDO DE CASTILLA CLASSIC OLD MEDIUM AMONTILLADO 750ml. Antique Amontillado, com mais tempo de solera, o que lhe confere um caráter único e muito mais complexidade. Antique Amontillado, com mais tempo de solera, o que lhe confere um caráter único e muito mais complexidade.

Fonte: O próprio autor

Na figura 14 é resultado o resultado do pré-processamento do texto.

Figura 14: Cenário 1 após a fase de pré-limpeza

vinho espanhol fernando castilla classic dry sec fino premium cream classic old medium amontillado 750ml vinho espanhol fernando castilla classic dry sec fino 750ml um classico jerez fino aromas frutas secas inconfundivel toque pungente conferido pelas leveduras flor mesclado um sutil toque oxidacao seco boa aciz muito equilibrado tem sabor marcante longa persistencia acompanha bem tapas espanhois frios alem amendoas secas levemente tostadas vinho espanhol fernando castilla premium cream 750ml vinho licadamente doce aromas frutas secas especiarias carvalho tostado macio concentrado tem longa persistencia sutil retro olfato excelente sobremesas bolo frutas secas nozes avela vinho espanhol fernando castilla classic old medium amontillado 750ml antique amontillado mais tempo solera lhe confere um carater unico muito mais complexida antique amontillado mais tempo solera lhe confere um carater unico muito mais complexida

Fonte: O próprio autor

O cenário 2 contêm 285 palavras referente ao tema bebida: Na figura xx é ilustrado o texto antes do pré-processamento

A figura 15 demonstra o texto do cenário 1 sem tratamento

Figura 15: Cenário 2 sem tratamento

O Mega Blend 2013 da Horal é uma Geuze feita em comemoração ao Toer de Geuze 2013, com blend de 9 produtores de lambic. São eles: 3 Fonteinen, Boon, Timmermans, Oud Beersel, Lindemans, De Troch, De Cam, Hanssens e Tilquin. Nota 99 no RateBeer, o maior ranking de cervejas do mundo A cerveja O "Tour da Geuze" ou "Toer de Geuze" (em flamengo), é um passeio que acontece a cada 2 anos na Bélgica onde é feita a visitação a alguns produtores de lambic, o estilo mais antigo de cerveja do mundo. Foi organizado pela primeira vez em 1997, onde as cervejaria Boon, De Cam, De Troch, 3 Fonteinen, Lindemans e Timmermans abriram suas portas para o público.

A organização deste passeio é feita pelo HORAL(Hoge Raad voor Ambachtelijke Lambikbieren), que trabalha como uma organização dos produtores artesanais de lambic.
Desde 2009, o HORAL resolveu criar um blend especial de lambics para comemoração deste evento, nascendo assim o Horal's Oude Geuze Mega Blend.
O blender é responsável por criar um blend perfeito, onde tenha a quantidade correta de lambic jovens e antigas, para que possa fazer a carbonatação corretamente.
Cerveja Belga
O Mega Blend de 2013 foi feito com um blend entre 9 lambics diferentes e maturado por cerca de 9 meses nas adegas controladas, para carbonatação natural e refinamento da bebida.
Produzida em garrafas de champagne de 750ml com rolha.
Uma cerveja que tem potencial de guarda de até 20 anos.
Aroma
O aroma traz notas ácidas, limão, brettanomyces e esteres frutais.
Sabor
No paladar mostra-se extremamente balanceada, com acidez marcante, esteres frutais, final seco e carbonatação frisante, tradicional das melhores Geuzes que existem.
Graduação alcoólica 7%
Melhor apreciada se servida entre 12ºC e 14ºC
Disponível em garrafas de 750ml.

Fonte: O próprio autor

Resultado é apresentado na figura 16 do pré-processamento

Figura 16: Cenário 2 após a fase de pré-limpeza

mega blend 2013 horal geuze feita comemoracao toer geuze 2013 blend 9 produtores lambic sao 3 fonteinen boon timmermans oud beersel lindemans troch cam hanssens tilquin nota 99 ratebeer maior ranking cervejas mundo a cerveja tour geuze toer geuze flamengo passeio acontece cada 2 anos belgica onde feito visitacao alguns produtores lambic estilo mais antigo cerveja mundo foi organizado primeira vez 1997 onde cervejaria boon cam troch 3 fonteinen lindemans timmermans abriram portas publico

a organizacao deste passeio feita horalhoge raad voor ambachtelijke lambikbieren trabalha organizacao produtores artesanais lambic desde 2009 horal resolveu criar blend especial lambics comemoracao deste evento nascendo assim horals oude geuze mega blend blender responsável criar blend perfeito onde tenha quantidade correta lambic jovens antigas possa fazer carbonatacao corretamente cerveja belga mega blend 2013 foi feito blend entre 9 lambics diferentes maturado cerca 9 meses adegas controladas carbonatacao natural refinamento bebida produzida garrafas champagne 750ml rolha cerveja tem potencial guarda ate 20 anos aroma aroma traz notas acidas limao brettanomyces esteres frutais sabor paladar mostra extremamente balanceada acidez marcante esteres frutais final seco carbonatacao frisante tradicional das melhores geuzes existem graduacao alcoolica 7% melhor apreciada servida entre 12ºc 14ºc disponivel garrafas 750ml

Fonte: O próprio autor

Nessa fase de pré-limpeza, ocorre a limpeza do texto, com a finalidade de remover, fatores que possam causar interferência no processamento do texto. Outro ponto importante dessa fase é primeira técnica de diminuir a dimensionalidade de termos presentes no texto, através da remoção de *stopwords* e utilizando os *stemmer*.

5.2 Resultados do mapa de redução

O cenário 1 representado pela figura 13 contém 164 palavras referentes ao tema bebida. Após a execução da fase de pré-processamento será executado o processo de mapa de redução, contando a quantidade de termos que aparecem no texto.

Resultado com a contagem dos termos para o mapa de redução será apresentado na tabela 2 a seguir:

Tabela 2: Resultado do mapa de redução 1

Termo	Total	Termo	Total	Termo	Total	Termo	Total
750ml	4	Confere	3	lidicamente	1	sabor	1
Aciz	1	Cream	2	longa	2	sec	2
acompanha	1	Doce	1	macio	1	secas	4
Além	1	Dry	2	mais	4	seco	1
amendoas	1	equilibrado	1	marcante	1	sobremesas	1
amontillado	4	espanhol	2	medium	2	solera	2
Antique	2	especiarias	1	mesclado	1	sutil	2
Aromas	2	excelente	1	muito	3	tapas	1
Avela	1	fernando	4	nozes	1	tem	2
Boa	1	Fino	3	old	2	tempo	2
Bolo	1	Flor	1	olfato	1	toque	2
Caráter	2	Frio	1	oxidação	1	tostadas	1
Carvalho	1	Frutas	3	pelas	1	tostado	1
Castilla	4	inconfudivel	1	persistencia	2	um	4
Classic	5	Jerez	1	premium	2	unico	2
complexida	2	levedura	1	pungente	1	vinho	5
concentrado	1	levemente	1	retro	1		

O cenário 2 será representado pela figura 15, contém 284 palavras referentes ao tema bebida. Após a execução da fase de pré-processamento será executado o processo de mapa de redução, contando a quantidade de termos que aparecem no texto. O resultado desse mapa de redução será representado pela tabela 3 abaixo:

Tabela 3: Resultado do mapa de redução 2

Termo	Total	Termo	Total	Termo	Total	Termo	Total
12ºc	1	Boon	2	geuze	6	Portas	1
14ºc	1	brettanomyces	1	geuzes	1	Possa	1
1997	1	Cada	1	graduacao	1	potencial	1
2	1	Cam	2	guarda	1	Primeira	1
20	1	carbonatacao	3	hanssens	1	produtores	3

2009	1	Cerca	1	horal	3	produzida	1
2013	3	cerveja	6	horalhoge	1	Publico	1
3	2	champagne	1	jovens	1	quantidade	1
7%	1	comemoracao	2	lambic	6	Raad	1
750ml	2	controladas	1	lambikbieren	1	Ranking	1
9	3	correta	1	limao	1	ratebeer	1
99	1	corretamente	1	lindemans	2	refinamento	1
Abriram	1	Criar	2	maior	1	Resolveu	1
Acidas	1	Desde	1	mais	1	responsavel	1
Acidez	1	diferentes	1	marcante	1	Rolha	1
acontece	1	disponivel	1	maturado	1	Sabor	1
Adegas	1	Entre	2	mega	3	São	1
Alcoolica	1	especial	1	melhor	1	seco	1
Alguns	1	esteres	2	melhores	1	servida	1
ambachtelijke	1	Estilo	1	meses	1	Tem	1
Anos	2	Evento	1	mostra	1	tenha	1
Antigas	1	existem	1	mundo	2	tilquin	1
Antigo	1	extremamente	1	nascendo	1	timmermans	2
apreciada	1	Fazer	1	natural	1	toer	3
Aroma	2	Feita	4	nota	1	trabalha	1
artesanais	1	Final	1	notas	1	tradicional	1
Assim	1	flamengo	1	onde	3	Traz	1
Ate	1	Foi	2	organizacao	2	troch	2
balanceada	1	fonteinen	2	oud	2	Vez	1
Bebida	1	frisante	1	paladar	1	visitacao	1
Beersel	1	Frutais	2	passeio	2	voor	1
Blender	1	garrafas	2	perfeito	1		

Fonte: O próprio autor

Essa fase é executada pelo Apache Hadoop, executou-se a contagem de termos presentes texto, nela o processamento da informação ocorre por processamento distribuídos e é gerado vetor de termos por recorrência.

5.3 Resultado do treinamento

Neste cenário será realizada a fase de treino para três categorias, representadas na tabela 4, com uma quantidade variável de produtos cadastrados e já conhecidos para cada uma das categorias. Será categorizado um produto, utilizando o modelo de vetorial gerado nessa categorização:

Tabela 4 Categorias X Produtos

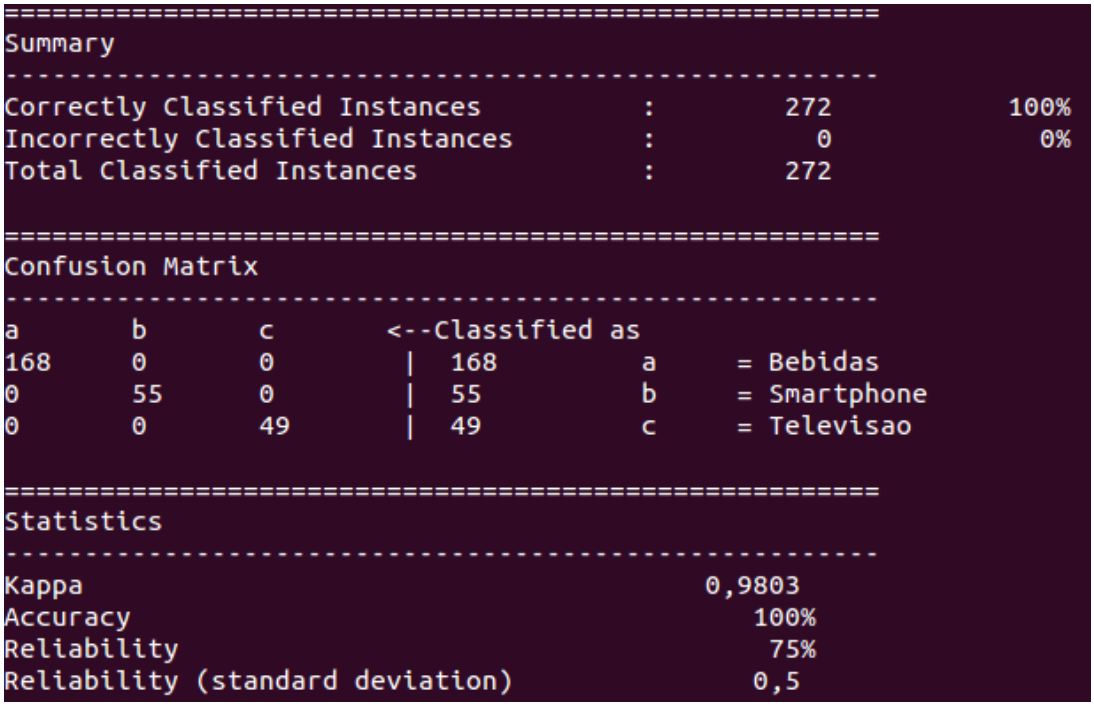
Categoria	Quantidade de Produtos
Smartphone	449
Televisão	550
Bebidas	1680

Para o processo de treinamento dos produtos ocorreu à divisão dos produtos em duas partes, sendo que a primeira parte cerca de 90% dos produtos foi responsável pelo treinamento do categorizador e o restante destinado à fase de testes do categorizador.

Como o processo de treino ocorre de forma implícita não é possível visualizar muitas informações do processo de treino, somente é possível visualizar as informações do tempo de treinamento, que nesta fase levou em média o tempo de 34.6 segundos.

Na fase de teste será possível visualizar a matriz confusão, que nada mais é que a representação dos produtos categorizados em conjunto com as categorizações sugeridas, Conforme representado na figura 17.

Figura 17 Resultado do teste categorização



```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :        272           100%
Incorrectly Classified Instances        :          0             0%
Total Classified Instances              :        272

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       c       <--Classified as
168     0       0        |  168         a     = Bebidas
0       55      0        |  55          b     = Smartphone
0       0       49       |  49          c     = Televisao

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0,9803
Accuracy                                     100%
Reliability                                   75%
Reliability (standard deviation)             0,5
```

Para a situação de treinamento que pode ser observada nas condições acima o processo de treinamento pode ser considerado excelente, já que o mesmo mostrar-se-á bem eficiente, e não houvera retorno de nenhuma categorização incorreta, considerando que somente seriam testados dez por cento (10%) de toda base de dados, conforme os valores demonstrados na figura 17.

5.4 Resultado da Categorização

No processo de categorização o produto utilizado é o elemento descrito abaixo pela figura 18:

Figura 18 Texto a ser categorizado

Maycas Sumaq Reserva Syrah 2012 (750 ml) Tinto frutado, macio, fresco e com notas de tabaco e chocolate amargo. Os vinhedos da Maycas del Limarí estão plantados na região norte do Chile, às portas do deserto do Atacama o que agrega a este vinho intensidade e complexidade.

Fonte: O próprio autor

No processo de categorização, do produto da figura 18 é cadastrado no banco de dados, no processo de categorização executado a categoria retornada, corretamente é a bebidas.

Como a base de dados utilizados, possui um treinamento ótimo para a representação da categoria de bebidas, o processo de categorização ocorre corretamente. Não sendo necessário realizar um novo treinamento, caso o produto fosse categorizado erroneamente seria necessário realizar uma nova fase de treinamento adicionando todos os produtos que tiveram problemas no processo de categorização automática. Com a finalidade de calibrar corretamente o categorizador.

5.5 Considerações Finais

Neste capitulo, foram demonstrados os resultados de cada fase de categorização. Na fase de pré-processamento, caracteriza-se como o primeiro passo para a redução da dimensionalidade do texto, removendo termos que não possuem relevância significativa ao conteúdo, e utilizando de técnicas de radicalização dos termos. No mapa de redução é demonstrado como os documentos podem ser representados, após fase de execução da primeira redução, onde os termos são estruturados seguindo o padrão de chave-valor, para a execução das demais fases. A fase de treinamento ocorre o processo de treinamento e geração de um modelo bayesiano na representação das categorias. E após todas essas etapas, finalmente ocorre o processo de categorização onde é feita uma análise, comparando o modelo bayesiano com o vetor de produtos retornando uma categoria mais provável para o produto.

A complexidade exigida no processo de entendimento das técnicas de categorização textual e os *frameworks* utilizados no processo de desenvolvimento. Trouxeram resultados satisfatórios em todos os resultados obtidos após o desenvolvimento da aplicação.

Conclusão

Com o volume de informações incorporadas ao dia- a- dia das pessoas, tais informações dificultam, ainda mais, a sua assimilação e organização, evidenciando o interesse em técnicas que facilitem a classificação delas. Com a finalidade de facilitar a recuperação dessas informações existe a pesquisa de técnicas que auxiliem esse processo de organização da informação que são bastante exploradas, apesar das mesmas apresentarem um alto grau de processamento no que tange ao agrupamento da informação.

Essas técnicas de classificação de textos podem ser aplicadas em uma diversidade de contextos, bem como, em *marketplace*, onde se encontram uma infinidade de produtos, que necessitam de um agrupamento de suas informações, em suas diversas categorias com a finalidade de facilitar a recuperação dessas, ou seja, exibir os produtos em categorias corretas, melhorando o processo de navegação de um usuário, ou até mesmo diminuir o custo de manutenção, onde seria necessária uma pessoa para realizar essa categorização dos produtos.

A técnica proposta será utilização da ferramenta de computação distribuídas no processamento dos textos dos produtos, devido ao alto grau de processamento necessário para manipular a informação em conjunto com ferramentas e técnicas de estruturação de textos para a estruturação dos diversos documentos e, assim, facilitar o processo de categorização dos produtos, com uso de ferramentas destinadas para a solução do problema.

No primeiro desenvolvimento do classificador automático, utilizando técnicas de similaridades de vetores serão realizados testes que demonstraram que a técnica proposta será ineficiente, apresentando logo nos primeiros testes uma baixa taxa de acerto, próximo a vinte por cento(20%), deixando claro que a proposta inicial para a solução problema será totalmente ineficiente. Na segunda tentativa, serão utilizadas técnicas de ferramenta artificiais para a solução do problema, que apresentarão uma taxa de acerto bem alta, próxima a cem por cento (100%), demonstrando que a técnica de categorização Naive Bayes será a melhor solução para o problema apresentado.

6.1 Trabalhos Futuros

Com base no trabalho produzido e nos resultados apresentados, há uma variedade de trabalhos que poderão ser realizados buscando melhorias, em pontos específicos ou generalizados.

Desta forma, caminhou-se além almejando futuras possibilidades de estudo e projetos de pesquisa nesta área de pesquisa sendo incorporados a esse trabalho, assim propõe--se:

- Análise de desempenho e eficiência de outras técnicas de categorização:
- Desenvolvimento de um categorizador utilizando técnicas de agrupamento de informações, ao invés de classificadores;
- Estudo e comparação com outras ferramentas de classificação textual;
- Uso de outras ferramentas de execução de processamento distribuídos;
- Avaliação do tempo de processamento da ferramenta de classificação de produtos, em diferentes ambientes de computação distribuída;
- Utilizar outras técnicas de tratamento de *stemmers* para o tratamento do texto e, avaliar seu impacto no processo de categorização.

Referências Bibliográficas

DEAN, J. GHEMAWAT, S: **MapReduce: Simplified Data Processing Large Clusters**, 2004.

ESPIRITO SANTO, A; **Categorização e análise de dados não estruturados: O caso dos debates parlamentares**, Universidade Nova Lisboa, 2009.

FURQUIM, L. O. C: **Agrupamento e categorização de documentos jurídicos**, Pontifícia Universidade Católica do Rio Grande do Sul, 2011.

GALHO, T. S.: **Categorização automática de documentos utilizando lógica difusa**, Universidade Luterana do Brasil, 2003.

GIACOMELLI, P.: **Apache Mahout cookbook**. Birmingham: Packet Publishing Ltd, 2013.

GOMES, F. T., PARDO, T. S.: **"Classificação e agrupamento de textos para processamento multi documento"**, Universidade de São Paulo, 2009

IBM developerWorks. **"Open Source Big Data for the Impatient, Part 1: Hadoop tutorial: Hello World with Java, Pig, Hive, Flume, Fuse, Oozie, and Sqoop with Informix, DB2, and MySQL"**. Disponível em: http://www.ibm.com/developerworks/data/library/techarticle/dm-1209hadoopbigdata/ Acesso em: março de 2014.

MACHINE LEARNING FOR LANGUAGE TOOLKIT: **"Mallet homepage"**. Disponível em: http://mallet.cs.umass.edu/index.php Acesso em: abril de 2014.

MANNING, C. D., RAGHAVAN, P., et al: **Introduction to Information Retrieval**, Cambridge University Press, 2008.

MEDEIROS, E. A: **Técnica de aprendizagem de máquina para a categorização de textos**, Universidade de Pernambuco, 2004.

OWEN, S. ROBI, Y : **Mahout in Action.** New York: Manning Publications, Co, 2011

PEREIRA, A.: **Mineração de dados distribuída usando Apache Mahout,** Universidade Federal de Santa Maria, 2010.

RIZZI C. B. et al: **"Categorização de textos por rede neural estudo de caso"**, Universidade Federal do Rio Grande do Sul, 2000

ROBERTSON, S.: **Understanding Inverse Document Frequency: On theoretical arguments for IDF**, Journal of Documentation 60 no. 5, pp 503–520, 2004

SANTOS, D. R. M**.: Recuperação da informação utilizando Apache Lucene e Wordnet**, Maringá, 2012.

THE APACHE SOFTWARE FOUDATION **:Welcome to Apache Hadoop!** Disponível em: http://hadoop.apache.org Acesso em: maio de 2014.

THE APACHE SOFTWARE FOUDATION **:Apache Mahout:: scalable machine-learning and data-mining library**. Disponível em: http://mahout.apache.org/ Acesso em: abril de 2014.

VERNNER, J.: **Pro Hadoop**, Apress 2009.

WIVES, L. K.: **Um estudo sobre agrupamento de documentos textuais em processamento de informações não estruturadas usando técnicas de clustering**, Universidade Federal do Rio Grande do Sul, 1999.

ZADEH, L.: **Fuzzy sets**, Information Control 8, 338–353, 1965.