CENTRO UNIVERSITÁRIO SENAC

Fernando Divanor Santana Alves
Jandi Jesus Santos

# Análise de Opinião em Ambiente Microblog

São Paulo
2010

FERNANDO DIVANOR SANTANA ALVES
JANDI JESUS SANTOS

# Análise de Opinião em Ambiente Microblog

Trabalho de conclusão de curso apresentado ao Centro Universitário SENAC – Campus Santo Amaro, como exigência parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Fabrício J. Barth

Co-Orientador: Prof. Dr. Orlando Rodrigues Jr.

São Paulo
2010

Alves, Fernando Divanor Santana
Santos, Jandi Jesus

Análise de Opinião em Ambiente Microblog / Fernando Divanor Santana Alves e Jandi Jesus Santos – São Paulo, 2010
41 fls, il. Color.; 31 cm

Orientadores: Prof. Dr. Fabrício J. Barth
Prof. Dr. Orlando Rodrigues Jr.

Trabalho de Conclusão de Curso – Centro Universitário SENAC – Campus Santo Amaro, São Paulo, 2010

**Alunos:** FERNANDO DIVANOR SANTANA ALVES e JANDI JESUS SANTOS

**Título:** Análise de Opinião em Ambiente Microblog.

Trabalho de Conclusão de Curso apresentado ao Centro Universitário SENAC – Campus Santo Amaro, como exigência parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Fabrício J. Barth

Co-Orientador: Prof. Dr. Orlando Rodrigues Jr.

A banca examinadora dos Trabalhos de Conclusão em sessão pública realizada Em 16/Dezembro/2010 considerou o(s) candidato(s):

( ) aprovado ( ) reprovado

1) Examinador (a):_____

2) Examinador (a):_____

3) Presidente: _____

# DEDICATÓRIA

Dedico esse trabalho a minha esposa Jacira, que sempre me incentivou na busca pelo conhecimento e esteve ao meu lado me apoiando e dando forças nos momentos mais difíceis da minha vida acadêmica.
(Jandi)

Dedico este trabalho a meus familiares, que sempre me deram apoio quando o cansaço me desestimulava
(Fernando)

# AGRADECIMENTOS

A Deus sem o qual não seria possível a realização desse trabalho.

Aos nossos orientadores Prof. Dr. Fabrício J. Barth e Prof. Dr. Orlando Rodrigues Jr. por acreditar em nossa capacidade, pela motivação e pela paciência em nos orientar.

Aos nossos familiares e amigos pelo apoio e compreensão nessa fase tão importante de nossas vidas.

E a todos aqueles que contribuíram indiretamente para a realização desse trabalho.

*Ele não sabia que era impossível. Foi lá e fez.*

Jean Cocteau

# RESUMO

A posposta deste trabalho é avaliar o uso de técnicas de *Data Mining* para capturar o sentimento predominante de textos escritos por usuários de ambientes microblog de forma automatizada. Para realizar tal tarefa foi desenvolvido um sistema capaz de capturar e classificar mensagens geradas no microblog *Twitter.*

O classificador utilizado faz uso da técnica de Naive Bayes que é um modelo estatístico amplamente utilizado na tarefa de classificação de textos. Os resultados encontrados neste trabalho sugerem que o algoritmo de classificação realiza a obtenção do sentimento das mensagens com taxas de assertividade aceitáveis, em média de 80%. Também se concluiu que o algoritmo tem melhor desempenho quando usado para classificar textos nas classes positiva e negativas e que ao incluir uma terceira classe, a neutra, o desempenho dos resultados é afetado negativamente.

Palavras chave: Mineração de Opinião; Classificação de Textos; *Twitter*, Classificação de Sentimento.

# ABSTRACT

The purpose of this research is to evaluate the possibility of using techniques to capture the sentiment or the predominant opinions of the users of social networks about a determined subject in an automated way. To perform such work a system capable of capturing and analyzing messages generated from Twitter was developed.

The classifier utilized makes use of the Naive Bayes technique which is a statistical model widely used in the classification of texts. The results analyzed show that classification algorithms perform well getting the sentiment of messages with acceptable rates of assertiveness, an average of 80%. It concludes that the algorithm has a better performance when used to classify texts into two classes, positive and negative. It was also conclude that when include a third class, the neutral, the performance of the results are negatively affected.



Keywords: Opinion Mining; Text Classification; Twitter; Sentiment Classification

## LISTA DE ILUSTRAÇÕES

## LISTA DE TABELAS

# SUMÁRIO

# 1 Introdução

Informações podem ser armazenadas de diversas maneiras, sendo que a maneira mais clássica de se realizar tal armazenamento se dá através da palavra, seja ela escrita ou impressa. O grande problema, é que a acessibilidade as informações armazenadas desta maneira, ocorrem de forma lenta e com desempenho pouco satisfatório (MANDEL, SIMON, & DELYRA, 1997).

O acesso a internet, computadores e dispositivos moveis que antes estavam apenas no âmbito das grandes instituições, hoje fazem parte do nosso cotidiano. A disponibilidade de crescentes coleções de textos no ambiente web, tornou inerente a vida das pessoas a interação com a rede, seja gerando ou fazendo uso de tais coleções de textos.

O advento dos computadores e a invenção de diversos sistemas de recuperação de informação tornam a capacidade de se armazenar e manipular dados, cada vez mais necessária, uma vez que hoje a *World Wide Web* é o maior repositório de dados do mundo (CHAKRABARTI, 2003).

Surge então a necessidade de se obter informações de forma automatizada devido à dificuldade implícita de se realizar tais tarefas de forma manual.Esta tarefa não é apenas tecnicamente desafiadora devido à necessidade de se utilizar técnicas de *Web Data Mining* ou mineração de dados na *Web* (KOSALA, 2000), mas é também muito útil na prática, um exemplo comum que possui crescentes quantidades de dados são os ambientes microblog.

As pessoas utilizam ambientes microblog para divulgar o que pensam sobre determinado assunto, para apoiar uma causa, criticar atitudes, divulgar eventos, enfim, manifestar opiniões.

A disponibilidade de opiniões obtidas através de técnicas de *Web Data Mining*, pode ser usada por empresas que desejam estreitar o relacionamento com seus clientes. Por outro lado, um cliente insatisfeito que possui uma grande quantidade de seguidores, pode fazer *marketing* negativo ao expressar sua insatisfação nesse ambiente.

Portanto, a fim de tornar mais eficiente o monitoramento de opiniões de usuários de microblogs, esse trabalho propõe a aplicação de classificação automatizada do sentimento contido em mensagens escritas em tais ambientes.

## 1.1 Microblogs e o *Twitter*

Resumidamente um microblog pode ser definido como uma mistura de blog com rede social, e mensagens instantâneas. O microblog tem em comum com o blog a idéia de atualizações em ordem cronológica inversa, permitem comentários de outros usuários, réplica de postagens de outros usuários, *trackbacks* etc.

Peculiaridades, como possuir tamanho reduzido para as postagens, apenas 140 caracteres proporciona dinamismo no conteúdo desses ambientes. A possibilidade de atualização através de dispositivos móveis como celulares, smartfones etc. permite que os microblogs sejam mais ágeis na cobertura de acontecimentos do que meios de comunicação tradicionais como a televisão.

O *Twitter*[1] é um ambiente microblog fundado em 2006 pela Obvious em São Francisco, foi lançado ao público em 13 de julho do mesmo ano. Essa ferramenta possibilita postagem de mensagens chamadas de *tweets* possuindo não mais que 140 caracteres por mensagem. As postagens podem ser feitas por SMS, por *instant messages*, por internet móvel ou por aplicativos construídos utilizando-se de uma API disponibilizada pelo próprio *Twitter*.

As mensagens postadas pelo usuário do *Twitter* ficam disponíveis em seu perfil público e são enviadas a seus seguidores. No contexto do *Twitter* um seguidor é o usuário que deseja receber as mensagens postadas por qualquer outro usuário, assim, quando o usuário seguido posta uma mensagem no *Twitter* a mensagem postada é encaminhada a todos seus seguidores. O usuário que recebe a mensagem postada tem a opção de responder o autor da mensagem ou utilizar um recurso chamado de *retweet* que consiste em postar em seu perfil a mensagem recebida referenciando seu autor, ou seja, quando se utiliza o *retweet*, “twitar-se” algo que alguém “twittou”, com isso a mensagem alcança todos os

[1] *http://twitter.com/about*

seguidores através da interconexão dos usuários, essa é a forma com que o *Twitter* faz a disseminação de informação.

O site começou com o slogan “O que você está fazendo” com a intenção de estimular seus usuários a postar tudo que está fazendo, o tempo inteiro, mas foi mudado para: “O que está acontecendo ao seu redor”, para estimular seus usuários a fazer postagens que reflitam o que está acontecendo ao redor do mundo.

Por se tratar de postagens curtas, é comum encontrar links que direcionam o usuário a outra fonte de informação, é comum também encontrar palavras escritas gramaticamente incorretas com a finalidade de abreviação, podendo assim inserir mais informação em 140 caracteres.

## 1.2 Objetivo Geral

O objetivo geral deste trabalho é avaliar a possibilidade de se utilizar técnicas de *Data Mining* para extração automatizada do sentimento (positivo, negativo ou neutro) de um tweet.

Os objetivos específicos para a realização deste trabalho são:

- Realizar a coleta e o armazenamento de dados para realização de testes e validações;
- Realizar pré-processamento dos dados a fim de avaliar o desempenho do modelo para diferentes dados com diferentes características;
- Implementar um algoritmo que possibilite a classificação de textos curtos encontrados em ambientes microblog.
- Analisar os resultados a fim de saber se a técnica escolhida é eficiente na classificação de textos com as características dos textos encontrados no domínio dos microblogs.

.

## 1.3 Justificativa

As pessoas utilizam os ambientes microblog para divulgar o que pensam sobre determinado assunto, para apoiar uma causa, criticar atitudes, divulgar eventos, enfim, manifestar opiniões. Esse comportamento pode ser benéfico para as empresas estreitarem o relacionamento com seus clientes, mas por outro lado, um cliente insatisfeito que possui uma grande quantidade de seguidores, pode fazer um *marketing* negativo considerável ao expressar sua insatisfação nesse ambiente. Essas grandes quantidades de opiniões publicadas diariamente se tornam interessantes para empresas, pois tem a capacidade de influenciar usuários que utilizam os ambientes microblog.

O *Twitter* possui mais de 105 milhões de usuários cadastrados, com 300 mil novos perfis criados a cada dia, recebe 180 milhões de visitantes únicos a cada mês e mais de 55 milhões de mensagens publicadas diariamente. Esses dados foram apresentados por Biz Stone (co-fundador do *Twitter*) durante a primeira *Twitter´s Chirp* (conferência do *Twitter* para desenvolvedores, realizada nos dias 14 e 15 de abril de 2010 em São Francisco (EUA)).

Olhando isso de uma forma mais apurada, podemos propor o seguinte questionamento: como empresas e pessoas podem fazer uso dessa quantidade de opiniões para obter informações que possam levá-las a desenvolver novas estratégias de *marketing*?

Hoje em dia existem ferramentas que fazem o monitoramento de entidades em redes sociais – inclusive no *Twitter*. Um exemplo é site Scup[2], essa ferramenta faz buscas de mensagens em redes sociais por palavras chave, e permite que o usuário classifique as mensagens encontradas em positivas, negativas ou neutras, para que a partir dessa classificação possa apresentar estatísticas sobre o monitoramento (SCUP). Porém, a tarefa de classificar manualmente as mensagens do Twitter se torna humanamente impossível dado o grande volume de mensagens postadas diariamente.

Por tanto, esse trabalho propõe aplicar análise de sentimento ao monitoramento de mensagens do *Twitter*, a fim de classificar as mensagens que

[2] http://www.scup.com.br

as pessoas escrevem sobre determinada entidade (i.e., marca, empresa, pessoa, produto etc.) em positivo, negativo ou neutro. Com sua concretização pode-se tornar mais eficiente o monitoramento de mensagens no ambiente de microblog *Twitter*.

# 2 Fundamentação Teórica

Para realizar tarefas relacionadas a esse estudo as principais abordagens que poderíamos usar são: Aprendizado de Máquina, Processamento de Linguagem Natural (PLN), Estatística Computacional (IC), Recuperação de Informação (RI), Ciência Cognitiva, Mineração de Textos e Mineração na Web.

## 2.1 Classificação de Sentimento

A classificação de sentimento ou Sentment Classification (PANG, LEE AND VAITHYANATHAN 2008), é o campo de Web Data Mining cujo objetivo é definir em um determinado texto, a orientação semântica ou orientação de opinião sobre determinada entidade, dizendo se aquele determinado texto expressa uma opinião positiva, negativa, ou neutra. Isto difere dos trabalhos de categorização de onde o objetivo é organizar e ordenar documentos de acordo com seu assunto principal. Segundo PANG (2008) alguns estudos sugerem que classificação de sentimento é mais difícil de ser realizada do que classificação de tópicos, o que torna a tarefa mais desafiadora.

A maior parte da pesquisa de análise de sentimento está baseada em duas linhas de estudo distintas no que concerne a maneira como essa extração é feita, mas similares em seu objetivo que é extrair sentimentos de textos.

## 2.2 Classificação Baseada em Frases de Sentimento

Este método realiza a classificação baseando-se no sentimento positivo ou negativo das palavras, do texto que está sendo avaliado, além de fazer uso da técnica de processamento de linguagem natural chamada *Part-of-speech* (POS) *tagging* que é o processo de marcar as palavras de um texto como correspondendo a um determinado rótulo.

O método baseia-se na definição da palavra, bem como seu contexto, isto é, relacionamento com palavras adjacentes e relacionadas em uma frase ou parágrafo, e a identificação de palavras como substantivos, verbos, adjetivos e advérbios.

## 2.3 Métodos de Classificação de Textos

Uma abordagem simples, mas muito eficiente para se extrair o sentimento de um texto é tratar o problema como um tarefa de classificação de texto, e então aplicar algum método de classificação como, por exemplo: Naive Bayesian e SVM (ALEC, RICHA & HUANG, 2009).

## 2.4 Escolha da Técnica de Classificação

Como o objetivo deste trabalho é realizar extração de sentimento em mensagens de ambientes de microblog, especificamente o *Twitter*, algumas peculiaridades do ambiente influenciaram na decisão de qual técnica seria mais apropriada.

No *Twitter* usuários podem postar mensagens de diferentes tipos de mídia, como por exemplo, celulares. Isto faz com que a taxa de erros gramaticais seja mais alta que nos demais domínios. Neste ambiente, a média de uso de gírias é muito alta.

Outro fator importante e que deve ser considerado, é que como dito previamente no *Twitter* as postagens são limitadas a 140 caracteres o que deixa pouco espaço para construções gramaticais mais complexas.

As peculiaridades descritas desqualificam o uso métodos de extração de sentimento que tem como fundamento o uso de processamento de linguagem natural. Sendo assim métodos de aprendizado de máquina é a escolha mais apropriada uma vez que não leva em conta se palavras estão escritos

corretamente, mas sim com qual freqüência esta palavra está associada a uma determinada classe.

Alguns trabalhos anteriores (PANG, LEE AND VAITHYANATHAN 2008) e (ALEC, RICHA & HUANG, 2009), realizados para língua inglesa, mostraram que é possível alcançar precisão em torno de 80% na classificação de sentimento usando os principais métodos de aprendizado de maquina. Nestes mesmos trabalhos percebe-se a eficiência de Naive Bayes que apesar de ser mais simples, com relação à implementação, apresentou taxas de acertos equivalentes a métodos mais complexos como SVM por exemplo.

## 2.5 Naive Bayesian Classification

Aprendizado supervisionado pode ser estudado de um ponto de vista probabilístico. Logo a tarefa de classificação pode ser considerada estimando a probabilidade a "*posterior*" da classe baseando-se em um conjunto de dados previamente classificado i. e. Conjunto de treinamento.

Há algumas versões de métodos classificadores Bayesianos, e em uma definição simplificada desta abordagem está fundamentada na teoria da probabilidade condicional (DUDA et.al. 2000). Se assumirmos que a probabilidade condicional de que palavras específicas aparecem em um determinado texto dado que o mesmo está associado a uma determinada classe, podemos então usar a técnica para calcular as probabilidades de um novo documento estar em cada uma das classes listadas, atribuindo para o mesmo a classe que possua maior probabilidade (LEWIS & RINGUETTE, 1994).

Apesar de sua simplicidade estudos comprovam que o algoritmo de aprendizagem supervisionada de Bayes tem se mostrado bastante eficaz na tarefa de classificação de textos (MANNING & SCHUTZE, 1999). Essa é uma das abordagens mais utilizadas na tarefa de classificação de textos (MITCHELL, 1997). Nesse estudo cada um dos exemplos (mensagens) do conjunto de treinamento, é representado como uma série de atributos os quais indicam a presença ou não dos termos (a1, a2,..., an), e o classificador tem como objetivo de realizar a atribuição da categoria que possua maior probabilidade para cada

mensagem através de uma função cujo retorno é um valor (classe) pertencente a um conjunto finito.

O algoritmo de aprendizagem supervisionado de Bayes tem como fundamento a suposição simplificada de que o conjunto de atributos dos exemplos de entrada ou treinamento é condicionalmente independente dado o valor final da função de saída. Logo este classificador assume que a probabilidade de ocorrência de uma conjunção de atributos em algum exemplo é igual ao produto das probabilidades da ocorrência de cada atributo isoladamente.

$$VNB = \arg\max_{vj \in V} P(a1, a2, \ldots, an/vj).P(vj) = \arg\max P(vj) . \prod_{i} P(ai/vj) \quad (1)$$

Onde:

- VNB é a hipótese (classe) final atribuída ao texto;
- vj é cada um dos possíveis valores (classes) que fazem parte de V;
- P(a1, a2,..., an/vj) é a probabilidade de ocorrência do conjunto de evidências dada a ocorrência da hipótese (classe);
- P(vj) é a probabilidade inicial de ocorrência de cada hipótese (classe);
- P(ai/vj) é a probabilidade de ocorrência de cada evidência dada à ocorrência de uma hipótese (classe).

Uma diferença entre o algoritmo de Bayes e as demais técnicas de aprendizado de máquina consiste no fato de que não há uma busca explicita no espaço das hipóteses, mas sim a definição de uma única hipótese contabilizando a freqüência das muitas combinações dos dados pertencentes ao conjunto de treinamento. Há diversas variações de algoritmos para realizar esse tipo de classificação, sendo que para o de Bayes existem dois subtipos o *multinomial model* e o *multivariate model*.

No *multinomial model* a probabilidade do termo T pertencer à classe C é calculada pela equação 2.

$$P(T/C) = \frac{\text{freqüência do termo T nos documentos classificados na categoria C}}{\text{freqüência do termo T em todos documentos classificados}} \quad (2)$$

Sendo A uma mensagem, representada por um vetor composto de n termos A = (t1,..., tn), e uma classe Ci. A probabilidade da mensagem A pertencer a i-ésima classe Ci é dada através do cálculo do produto dos n termos t dessa mensagem pertencerem a essa classe C, descrito na seguinte equação:

$$P(A/C_i) = \prod_{j=1}^{j=n} (tj \mid Ci) \quad (3)$$

Por se tratar de uma produtória não e muito difícil perceber que a inexistência de algum atributo, ou seja, quando a probabilidade de algum termo pertencer a uma determinada classe for zero, a probabilidade da mensagem pertencer a esta classe também será zero. Para contornar esse tipo de situação pode-se adicionar uma constante maior que zero a probabilidade de cada termo (JACKSON & MOULINIER, 2002), aplicando um fator de correção ou simplesmente ignorando atributos inexistentes não os levando em consideração no calculo das probabilidades.

No caso do *multivariate model* um vetor de componentes binários indicam, para cada termo do vocabulário, se ele acontece ou não no documento. Não é computada qual a freqüência com que o termo aparece em mensagens novas, apenas sua presença ou ausência.

A maior parte das suposições feitas pelo método de Naive Bayes é violada na prática. Por exemplo, palavras em um documento não são independentes umas das outras. Apesar de tais violações pesquisadores tem demonstrado que o método Naive Bayes de aprendizado tem alcançado resultados com níveis de precisão próximo aos dos algoritmos como SVM por exemplo. (PANG, LEE AND VAITHYANATHAN 2008)

Naive Bayes também é muito eficiente, pois realiza a leitura dos dados do conjunto de treinamento apenas uma vez, para com isso estimar todas as probabilidades requeridas na classificação. O modelo pode ser usado de forma incremental, além do fato de que pode ser alterado facilmente com a inclusão de novos dados uma vez que probabilidades podem ser convenientemente revisadas.

# 3 Projeto

Por se tratar de uma área recente de pesquisa, ainda não se sabia qual técnica seria mais adequada ao contexto de microblog. Portanto, a primeira fase do projeto foi o estudo sobre técnicas de análise de sentimento mais adequadas para o contexto do trabalho.

## 3.1 Visão Geral

A hipótese que esse projeto visa comprovar é que a aplicação do método de classificação de Bayes é eficiente para extrair o sentimento predominante de mensagens postadas no *Twitter*. Para comprovação da hipótese sugerida foram realizadas as etapas descritas nas subseções:

## 3.2 Escolha das Entidades Monitoradas

O projeto abrange apenas dados coletados na língua portuguesa sendo que para efeito de estudo foram coletada mensagens relacionadas aos presidenciáveis nas eleições de 2010 e o atual presidente da república.

A distribuição dos dados entre os candidatos não foi verificada, no sentindo de que não foi levada em consideração se mensagens sobre um determinado presidenciável possuía mais freqüência com relação aos outros. O intuito é apenas a classificação das mensagens e não determinar a aceitação de um ou outro candidato.

Outro fator importante é que apesar de ser possível restringir a existência de *retweet*s durante a coleta de dados, Foi preferível manter os mesmos, pois a existência de *retweet*s reflete o ambiente real.

A tabela 1 possui as entidades monitoradas e as palavras-chave utilizadas na coleta dos dados.

**Tabela 1 - Entidades monitoradas.**

| Entidade | Palavra chave para busca |
|---|---|
| Dilma Rousseff | dilma |
| Luiz Inácio Lula da Silva | lula |
| José Serra | serra, josé serra |
| Marina Silva | marina silva |

## 3.3 Coleta e Armazenamento de Dados

É imprescindível que os dados sejam coletados no *Twitter* sejam salvos em uma base de dados, para possibilitar a realização de todo o processamento necessário ao estudo proposto.

A coleta das mensagens foi realizada a partir da implementação de um componente de *software* (*collector*) que faz acesso a API do *Twitter* trazendo as mensagens da *time line* do *Twitter*. A API não permite acesso ao histórico de mensagens, por essa razão coletamos as mensagens à medida que foram postadas. A limitação imposta pela API permitiu o retorno de um numero limitado de 16 *tweets* em cada consulta.

O monitoramento durou cerca de três dias rodando de forma contínua, onde foram coletadas e armazenadas mais de 14.000 mensagens.

Os tweets foram armazenados com algumas informações complementares a mensagem, como qual a entidade relacionada, sua data, seu Id etc. para posterior criação de uma ferramenta de monitoração de redes sociais.

## 3.4 Filtragem das Mensagens Coletadas

Nem todas as mensagens coletadas foram utilizadas nesse trabalho, algumas delas tinham características pouco interessantes ao estudo, por exemplo, mensagens contendo apenas um link onde a informação está de fato.

As mensagens utilizadas no estudo foram selecionadas uma a uma através do processo de classificação manual de sentimento realizada para posterior treinamento e validação da classificação automática gerada pelo algoritmo.

No contexto do projeto, não foi utilizado mensagens que pudessem confundir a pessoa que realizou a análise manual. São mensagens que fazem menção a mais de uma entidade, nessas mensagens pode-se falar mal de uma entidade comparando-a com outra entidade, isso aumenta a subjetividade da interpretação do sentimento da classe, pois a mensagem pode ser negativa em relação a uma entidade e positiva em relação à outra, isso que pode ocasionar erros na classificação manual.

Por essa razão ao realizar a análise manual, selecionamos mensagens que façam menção a apenas uma entidade por mensagem.

## 3.5 Pré-processamento das mensagens

O pré-processamento dos textos pode ser subdividido nas seguintes partes:

- **Tokenização:** A Tokenização é o processo onde o texto tem seus termos separados, são utilizados: espaço em branco, tabulações e quebra de linha para delimitar a separação dos termos. Esse processo também remove caracteres especiais como: ! @ # $ % ¨ & * ( ) ^ ? { } [ ] + =, além de números e pontuação. Tudo isso visa deixar o texto o mais limpo possível. Além do processo descrito os termos são convertidos para letra minúscula a fim de assumir uma padronização nos termos tratados.
- **Remoção de *stopwords*:** É o processo de tratamento de dados que remove palavras de parada, tais como: a, e, dos, de, com, como, contra, contudo, cuja, dele etc. Essas palavras são consideradas sem relevância, em qualquer corpus aparecem com uma freqüência alta na maioria dos documentos, ou seja, agregam pouco ou nenhuma informação ao processo de classificação de textos.

- **Unigrama e Bigrama:** O pré-processamento unigrama considera o termo como sendo um único elemento de texto, ou seja, uma única palavra.

  O pré-processamento de bigrama considera o termo como sendo dois elementos de texto, ou seja, duas palavras. Quando utilizamos esse tipo de processamento o termo é gerado pela concatenação de uma palavra com a próxima palavra da mensagem separada por um espaço em branco, como no exemplo para a mensagem:

  “O Brasil irá vencer a copa do mundo.”

  A mensagem do exemplo gera os seguintes termos: o Brasil, Brasil irá, irá vencer, vencer a, a copa, copa do, do mundo.

- ***Stemming:*** É o pré-processamento utilizado para redução de um termo ao seu radical, removendo seus afixos. Nesta fase do pré-processamento todos os *tokens* são processados por um *stemmer* que é um algoritmo cujo objetivo é reduzir as palavras a uma forma comum de apresentação, conhecida como *stem* (ou radical), fundindo ou combinando as formas morfologicamente variantes de um termo (FRAKES; BAEZA-YATES, 1992). Com a utilização desse tipo de processamento deseja-se que os termos derivados do mesmo radical sejam considerados um único termo, tendo assim uma junção do número de termos que tem significados semelhantes.

  Nesse estudo o processo de lematização contribuiu com o agrupamento dos termos no algoritmo de Bayes uma vez que dois ou mais atributos podem possuir o mesmo radical.

## 3.6 Arquitetura do protótipo.

Nesta seção descrevemos o modelo de arquitetura adotado, a interoperabilidade de componentes, a distribuição dos módulos do sistema, e apresentaremos a função de cada um de seus componentes. Na figura 1 é apresentada uma visão geral da arquitetura do protótipo.

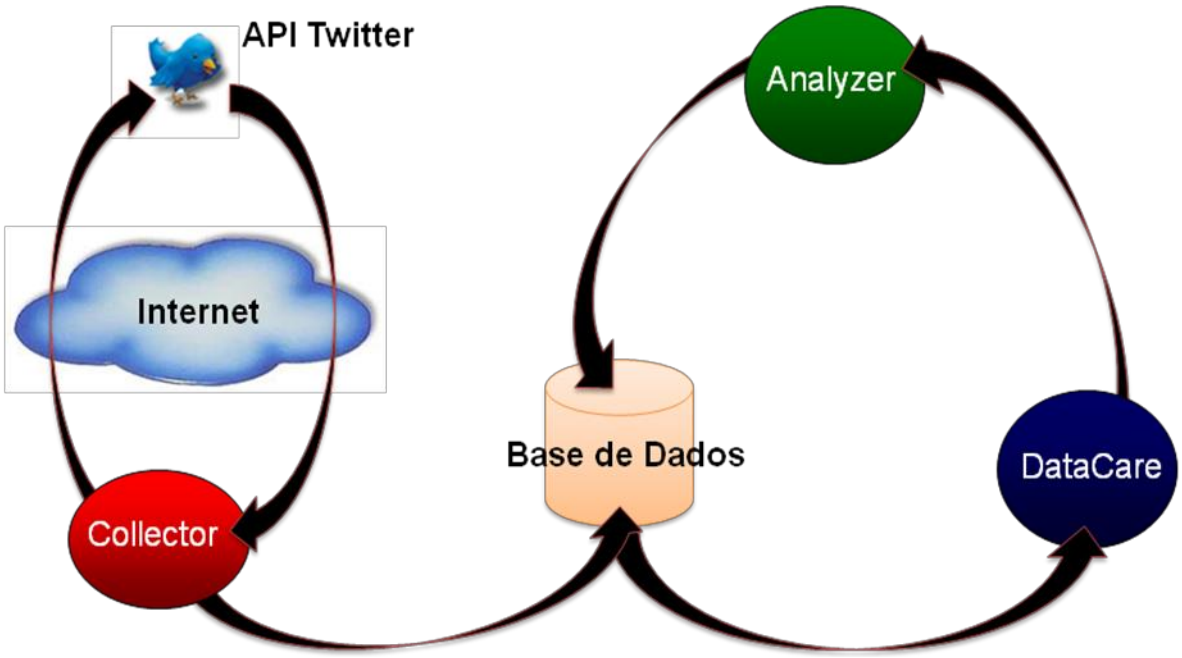


**Figura 1 – Visão geral da arquitetura do protótipo.**

### *3.6.1 Collector*

Esse componente é responsável pela coleta de mensagens no *Twitter*. O *Collector* utiliza o método search da API do *Twitter* para coletar mensagens buscando-as pelas palavras-chaves correspondentes a entidade monitorada.

Para cada entidade é criado uma *thread* independente, cada *thread* é responsável por monitorar e buscar mensagens de uma entidade.

As mensagens coletadas são armazenadas na base de dados para que sejam devidamente tratadas e analisadas. Esse componente se comunica com a API do *Twitter* e com o banco de dados da aplicação.

Ao realizar uma requisição a API d *Twitter* o coletor recebe um XML semelhante ao da figura 2.

```
</entry>
- <entry>
    <id>tag:search.twitter.com,2005:4961770171015168</id>
    <published>2010-11-17T18:19:12Z</published>
    <link type="text/html" href="http://twitter.com/raqmartinez/statuses/4961770171015168" rel="alternate" />
    <title>minha mae acabou de soltar um : QUÉ QUÉ ISSO ? tipo marina silva do @programapanico</title>
    <content type="html">minha mae acabou de soltar um : QUÉ QUÉ ISSO ? tipo <b>marina</b> <b>silva</b> do <a
      href="http://twitter.com/programapanico">@programapanico</a></content>
    <updated>2010-11-17T18:19:12Z</updated>
    <link type="image/png" href="http://static.twitter.com/images/default_profile_normal.png" rel="image" />
    <twitter:geo />
  - <twitter:metadata>
      <twitter:result_type>recent</twitter:result_type>
    </twitter:metadata>
    <twitter:source><a href="http://twitter.com/">web</a></twitter:source>
    <twitter:lang>pt</twitter:lang>
  - <author>
      <name>raqmartinez (Raquel Martinez)</name>
      <uri>http://twitter.com/raqmartinez</uri>
    </author>
  </entry>
- <entry>
```

**Figura 2 - XML recebido em uma requisição a API do *Twitter*.**

### *3.6.2 DataCare*

Esse componente tem a responsabilidade de realizar o pré-processamento das mensagens para que possam ser analisadas e classificadas. Remove palavras irrelevantes para análise, tais como: remoção de stopwords, links, números e caracteres especiais. Os resultados do tratamento são armazenados na base de dados para posterior análise. Esse componente se comunica diretamente com o banco de dados da aplicação.

O *DataCare* foi desenvolvido para ser capaz de realizar os seguintes tipos de tratamentos:

tokenização unigrama;

tokenização unigrama mais remoção de *stopwords*;

tokenização unigrama mais *stemming*;

tokenização unigrama mais remoção de *stopwords* mais *stemming*;

tokenização bigrama;

tokenização bigrama mais remoção de *stopwords*;

tokenização bigrama mais *stemming*;
tokenização bigrama mais remoção de *stopwords* + *stemming*;

Exemplo de tratamento:

**Mensagem original:** *RT @VEJA: 'Lula - o Filho do Brasil' será o filme que representará o Brasil no Oscar de 2011 | http://migre.me/1nLT5 via @radaronline*

**Mensagem tratada com tokenização unigrama mais *stemming* mais remoção e *stopwords*:** *veja lula filh brasil ser film represent brasil oscar via radaronline*

### *3.6.3 Analyzer*

Esse é o analisador de mensagens do *Twitter*, tem a responsabilidade de classificar mensagens em positiva, negativa ou neutra. Este é o principal componente do projeto, e onde é implementado o algoritmo de aprendizagem supervisionado de Bayes.

O analisador faz buscas na base de dados por mensagens tratadas que ainda não foram analisadas e classificadas. Antes de analisar a mensagem deve esperar até que o componente *DataCare* processe a mensagem.

Esse componente foi desenvolvido diretamente no banco de dados para ter um acesso mais rápido às informações necessárias à análise das mensagens, não tendo a responsabilidade de se comunicar com nenhum outro componente, visto que realiza a última fase do processo de classificação de mensagens.

### 3.6.4 Tecnologias Utilizadas

Como a nossa fonte de dados está disponível em uma ambiente web, é necessário o uso de uma linguagem que possua os recursos necessários para interagir com esse ambiente, realizar o tratamento de dados nos formatos

disponibilizados, como XML, Json etc. As tecnologias que utilizamos no processo formam:

- Linguagem de programação C#[3];
- Linguagem de banco de dados Transact-SQL[4];
- SGBD SQL Server 2005;
- Biblioteca Apache Lucene.

A linguagem de banco de dados foi selecionada visto que tem melhor integração com a linguagem de programação escolhida. Utilizada principalmente no desenvolvimento do núcleo do analisador, pois proporciona um melhor desempenho por ser implementada diretamente no banco de dados.

[3] http://msdn.microsoft.com/en-us/vcsharp/aa336809.aspx
[4] http://msdn.microsoft.com/pt-br/library/bb510741.aspx

# 4 Testes e Validação

Com objetivo de identificar o tipo de pré-processamento mais adequado para a aplicação do algoritmo e como se comporta o algoritmo quando diminuímos o número de classes a serem consideradas na análise, foram realizados oito testes variando o tipo de pré-processamento das mensagens para duas (positiva e negativa) e três classes (positiva negativa e neutra).

Para validar o desempenho do algoritmo proposto nesse estudo os testes foram realizados utilizando a técnica estatística de validação *Cross Validation*. Para comparação de desempenho dos testes utilizamos a medida de *acuracy* ou acurácia gerada a partir do resultado do *Cross Validation*.

Para comprovar a assertividade do modelo implementado, os testes visam comparar a classificação realizada manualmente com a classificação automatizada gerada pelo algoritmo.

## 4.1 Classificação Manual

O processo de classificação manual ou classificação humana consiste na análise do sentimento das mensagens pela perspectiva humana. Esse processo tem o objetivo de gerar o conjunto de treinamento do algoritmo de Bayes, assim como contribuir para validação dos resultados do mesmo.

Cerca de 3100 mensagens foram analisadas manualmente, porém nem todas foram utilizadas nos estudo. Grande parte das mensagens não tinha uma quantidade relevante de palavras para gerar o conjunto de treinamento rico em termos. Mensagens com essas características foram descartadas.

Foram descartadas ainda algumas mensagens consideradas interessantes ao estudo com a intenção de manter uma distribuição equilibrada entre as classes: positiva, negativa e neutra. Grande parte dessas mensagens pertencia à classe negativa.

Ao final do processo foi aproveitado um conjunto de 1950 mensagens distribuídas de forma equilibrada entre as classes.

As mensagens classificadas manualmente foram utilizadas para validação do modelo comparando-as com a classificação automática gerada pelo algoritmo de Naive Bayes.

## 4.2 *Cross Validation*

O *Cross Validation*, muitas vezes chamada de estimativa de rotação é uma técnica estatística de validação de algoritmos de aprendizagem utilizada para avaliar como os resultados de um algoritmo de aprendizagem se comporta ao receber conjuntos de dados independentes. (PAYAM REFAEILZADEH, LEI TANG, HUAN LIU). A utilização mais comum dessa técnica se da quando se quer avaliar o quanto um modelo preditivo irá funcionar na prática como é o nosso caso.

Uma das maneiras de utilizar o *Cross Validation* é o k-*fold Cross Validation* onde k é o número de subconjuntos de dados e número de iterações que será necessário para a execução da técnica (PAYAM REFAEILZADEH, LEI TANG, HUAN LIU, 2009). No nosso caso utilizaremos o *3-fold Cross Validation*.

O processo consiste em dividir o conjunto de dados que se deseja utilizar de forma aleatória, para validação do algoritmo em três subconjuntos de dados igualmente distribuídos. Após a divisão iniciam-se os testes submetendo um dos subconjuntos para que o algoritmo classifique (conjunto de teste), e os demais como conjunto de treinamento. Repete-se o processo até que todos os subconjuntos tenham sido utilizados como conjunto de teste e como conjunto de treinamento. O resultado desse processo é uma matriz, onde é apontada a média de acertos e erros de cada classe. A tabela 2 mostra a matriz de resultados onde constam os verdadeiros positivos e negativos, ou seja, a quantidade de mensagens que foram classificadas corretamente, e os falsos positivos e negativos, correspondentes a quantidade de mensagens classificadas de forma incorreta.

**Tabela 2 - Matriz de Resultados.**

| Classes | C1 | C2 |
|---|---|---|
| C1 | VP | FP |
| C2 | FN | VN |

## 4.3 Acurácia

Com base nos dados da tabela 2 podemos calcular a acurácia do algoritmo de acordo com a equação 4.

$$Acurácia = \frac{VP + VN}{FN + FP + VP + VN} \qquad (4)$$

Ou seja, a acurácia do algoritmo é equivalente ao percentual de acertos que o algoritmo alcançou.

## 4.4 Características do Conjunto de Dados para Testes

Para que seja possível realizar comparações entre os testes, características como distribuição equivalente da classificação manual entre as classes devem ser respeitadas em todos os testes. As únicas variações entre os tipos de testes são as características de pré-processamento e quantidade de classes consideradas na análise.

### 4.4.1 Conjuntos de Dados para Testes com Três Classes

Para os testes com três classes foi utilizado um conjunto de dados com 1950 mensagens com distribuição de 33% para cada classe, ou seja, as mensagens foram classificadas manualmente da seguinte forma: 650 positivas, 650 negativas e 650 neutras.

Para a execução do *Cross Validation* o conjunto de dados foi dividido aleatoriamente em três grupos de 650 mensagens, a distribuição entre as classes é apresentada na tabela 3.

**Tabela 3 - Distribuição de dados para *cross validation* com três classes.**

| Subconjunto/Classe | POSITIVAS | NEGATIVAS | NEUTRAS |
|---|---|---|---|
| **Subconjunto 1** | 229 | 210 | 211 |
| **Subconjunto 2** | 210 | 210 | 230 |
| **Subconjunto 3** | 211 | 230 | 209 |

### 4.4.2 Conjuntos de Dados para Testes com Duas Classes

Para os testes com duas classes foi utilizado um conjunto de dados com 1300 mensagens, esse conjunto de dados não possui o mesmo número de mensagens que o conjunto de dados utilizado nos testes com três classes (1950 mensagens). Isso porque foram retiradas 650 mensagens (de classe neutra) do conjunto de dados original para avaliar o desempenho do algoritmo com duas classes (positiva e negativa).

Porém foi respeitada a distribuição de mensagens entre as classes sendo 50% para mensagens classificadas como positivas e 50% para as classificadas como negativas, ou seja, 650 mensagens para cada classe.

Para a execução do *Cross Validation* o conjunto de dados foi dividido aleatoriamente em dois grupos de 433 mensagens e um de 434, a distribuição entre as classes é apresentada na tabela 4.

**Tabela 4 - Distribuição de dados para *cross validation* com duas classes.**

| Subconjunto/Classe | POSITIVAS | NEGATIVAS |
|---|---|---|
| **Subconjunto 1** | 205 | 228 |
| **Subconjunto 2** | 217 | 216 |
| **Subconjunto 3** | 228 | 206 |

## 4.5 Tipos de Testes

Os testes foram realizados de acordo com a tabela 5 aplicando o *Cross Validation* de acordo com as características dos conjuntos de dados descritos nas seções 4.4.1 e 4.4.2.

**Tabela 5 - Tipos de testes realizados.**

| Tipo de Pré-Processamento | Qtde. Classes | Teste ID |
|---|---|---|
| Tokenização Unigrama | 3 | **T1** |
| Tokenização Unigrama | 2 | **T2** |
| Tokenização Unigrama + Remoção de Stopwords | 3 | **T3** |
| Tokenização Unigrama + Remoção de Stopwords | 2 | **T4** |
| Tokenização Unigrama + Stemming | 3 | **T5** |
| Tokenização Unigrama + Stemming | 2 | **T6** |
| Tokenização Unigrama + Remoção de Stopwords + Stemming | 3 | **T7** |
| Tokenização Unigrama + Remoção de Stopwords + Stemming | 2 | **T8** |
| Tokenização Bigrama | 3 | **T9** |
| Tokenização Bigrama | 2 | **T10** |
| Tokenização Bigrama + Remoção de Stopwords | 3 | **T11** |
| Tokenização Bigrama + Remoção de Stopwords | 2 | **T12** |
| Tokenização Bigrama + Stemming | 3 | **T13** |
| Tokenização Bigrama + Stemming | 2 | **T14** |
| Tokenização Bigrama + Remoção de Stopwords + Stemming | 3 | **T15** |
| Tokenização Bigrama + Remoção de Stopwords + Stemming | 2 | **T16** |

# 5 Análise de Resultados

A tabela 6 apresenta os resultados dos testes, considerando três classes na análise.

**Tabela 6 - Resultados para análise com três classes.**

| Teste ID | Pré-Processamento | Acurácia | Desvio Padrão |
|---|---|---|---|
| T1 | Tokenização Unigrama | 55,85% | 3,63% |
| T3 | Tokenização Unigrama + Remoção de Stopwords | **56,46%** | 2,22% |
| T5 | Tokenização Unigrama + Stemming | 56,41% | 3,29% |
| T7 | Tokenização Unigrama + Remoção de Stopwords + Stemming | 55,08% | 2,28% |
| T9 | Tokenização Bigrama | 53,18% | 5,98% |
| T11 | Tokenização Bigrama + Remoção de Stopwords | 48,67% | 5,67% |
| T13 | Tokenização Bigrama + Stemming | 51,18% | 7,74% |
| T15 | Tokenização Bigrama + Remoção de Stopwords + Stemming | 48,36% | 5,23% |

Pode-se analisar que os resultados obtidos com três classes não chegaram a 57%. Nota se também que a variação de acurácia entre os vários tipos de pré-processamento não chegou a 8% com melhor desempenho apresentado no Teste T3 e pior desempenho no T15.

Isso mostra que a remoção de *stowords* e processamento de *stemming* não apresentam mudanças significativas no desempenho do modelo, podendo em casos como T15 piorar o desempenho do modelo quando combinados.

O processamento de bigrama também não apresentou melhoras significativas nos resultados da análise com três classes, nenhum dos testes realizados com pré-processamento de bigrama foi superior ao seu equivalente unigrama.

A tabela 7 apresenta os resultados dos testes apenas a classificação de mensagens positivas e negativas.

**Tabela 7 - Resultados para com duas classes.**

| Teste ID | Pré - Processamento | Acurácia | Desvio Padrão |
|---|---|---|---|
| T2 | Tokenização Unigrama | 79,38% | 2,15% |
| T4 | Tokenização Unigrama + Remoção de Stopwords | 76,85% | 3,78% |
| T6 | Tokenização Unigrama + Stemming | 79,15% | 1,61% |
| T8 | Tokenização Unigrama + Remoção de Stopwords + Stemming | 75,54% | 3,02% |
| T10 | Tokenização Bigrama | **82,31%** | 6,09% |
| T12 | Tokenização Bigrama + Remoção de Stopwords | 65,54% | 12,88% |
| T14 | Tokenização Bigrama + Stemming | 69,77% | 13,34% |
| T16 | Tokenização Bigrama + Remoção de Stopwords + Stemming | 63,97% | 31,18% |

Ao analisar os resultados obtidos com duas classes percebe-se que o melhor resultado chegou a quase 83%, pode-se também perceber que variação da acurácia passa de 18% com melhor desempenho no teste T10 e pior desempenho no teste T16, com quase 64% de acurácia.

O teste realizado com processamento de *stemming* apresentou melhor resultado que com a remoção de *stopwords*, porém a mudança não é muito significativa, chegando a quase 3%. Quando combinados causam queda no desempenho do algoritmo, como mostra a comparação entre os testes T8 e T6.

Os pré-processamentos de bigramas apresentaram pior desempenho que os pré-processamentos unigramas equivalentes, com exceção do T10 que mostrou o melhor desempenho de todo estudo.

## 5.1 Três Classes X Duas Classes

Pode-se perceber que se obtêm resultados melhores quando se considera apenas duas classes na análise do que quando se considera três classes. A maior diferença entre os testes equivalentes de duas e três classes chega a 29%.

É aceitável que o algoritmo lide melhor com duas classes do que três, visto que se considerarmos um chute ao acaso em uma das três classes as chances de erro é de 66% enquanto que a chance de acerto é e apenas 33%. Ao fazer a mesma análise com duas classes percebemos que as chances tanto de acerto quanto de erro são de 50%.

Outro fator que pode influenciar no desempenho é que as mensagens utilizadas nos teste com duas classes foram classificadas como positivas e negativas. Os termos existentes nessas classes são muitas vezes opostos, ou

seja, é mais improvável que um termo presente na classe positiva também esteja presente na classe negativa. Isso permite uma melhor separação dos dados e conseqüentemente melhor desempenho do modelo.

Diferente das características das classes positivas e negativas a classe neutra possui termos que podem se repetir com mais freqüência nas demais classes. Uma hipótese mais provável da obtenção de piores resultados com três classes, é que classe neutra cause ruído no modelo, pois sua característica diminui a separação dos dados.

### 5.1.1 Ruído da Classe Neutra

Para comprovar a hipótese levantada na seção 5.1 sugere-se a realização de um novo teste com duas classes, variando a combinação de classes a fim de medir o desempenho do algoritmo para comparar a diferença de desempenho gerada quando incluímos a classe neutra na análise.

Para realização do teste proposto foi escolhido a o pré-processamento T10 (Tokenização Bigrama), por apresentar o melhores resultados e maior diferença de desempenho quando comparado ao seu equivalente de três classes.

A execução dos testes foi realizada da mesma forma que os testes com duas classes anteriores. A tabela 8 mostra o desempenho de cada teste.

**Tabela 8 - Resultados de diferentes combinações de classes.**

| Tokenização Bigrama | Acurácia | Desvio Padrão |
|---|---|---|
| **Positiva e Negativa** | 81,85% | 4,87% |
| **Positiva e Neutra** | 64,63% | 10,48% |
| **Negativa e Neutra** | 58,44% | 4,47% |

Pode-se observar que mesmo com duas classes, quando a classe neutra está presente na análise o desempenho cai cerca de 17% quando combinada com a classe positiva e 23% quando combinado com a classe negativa.

Com os resultados obtidos podemos comprovar a hipótese levantada na sessão 5.1, é verdadeira, ou seja, a classe neutra causa ruído no modelo, pois

suas características contribuem para o aumento da ambigüidade entre os termos do conjunto de treinamento.

Seguindo esse raciocínio faz sentido pensar que quanto mais ambigüidade entre os termos existirem no conjunto de treinamento pior o desempenho do algoritmo.

Podemos perceber essas características quando fazemos uma comparação entre o desempenho do algoritmo e a quantidade de termos considerados ambíguos, como mostra gráfico da Figura 3.

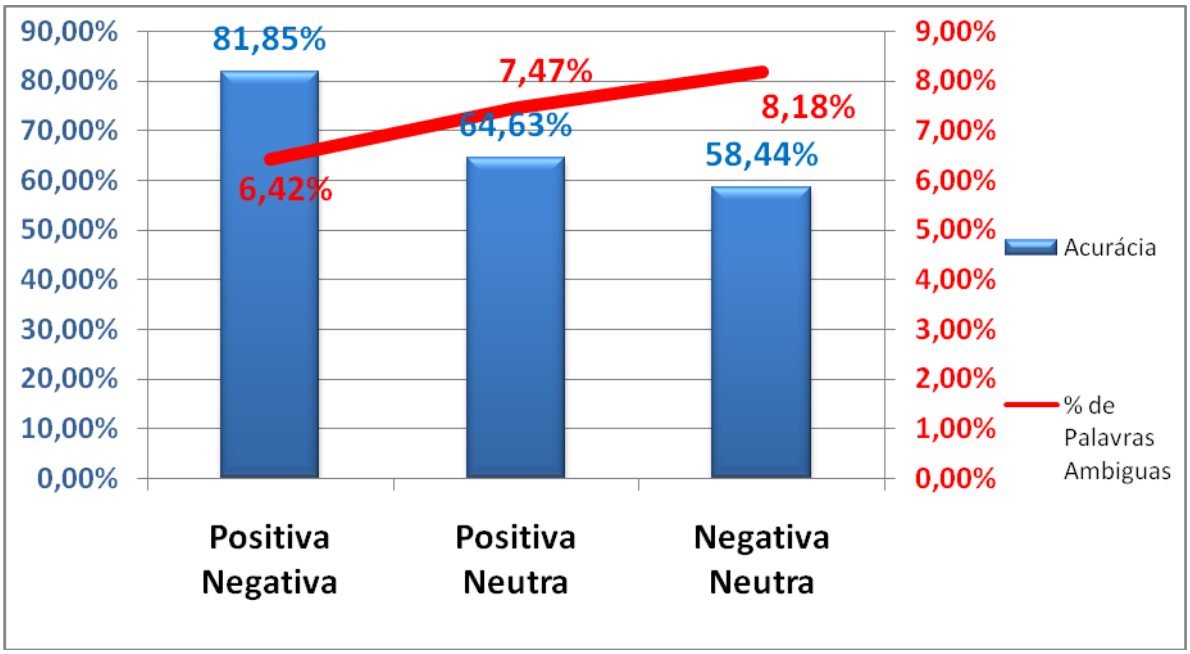


**Figura 3 - Relação entre ambigüidade da classificação dos termos e desempenho.**

O gráfico da figura 3 sugere que quanto mais palavras ambiguas existem no conjunto de treinamento pior é o desempenho do algoritmo. O diagrama da figura 4 mostra o ponto de intersecção de termos ambiguos entre os testes realizados.

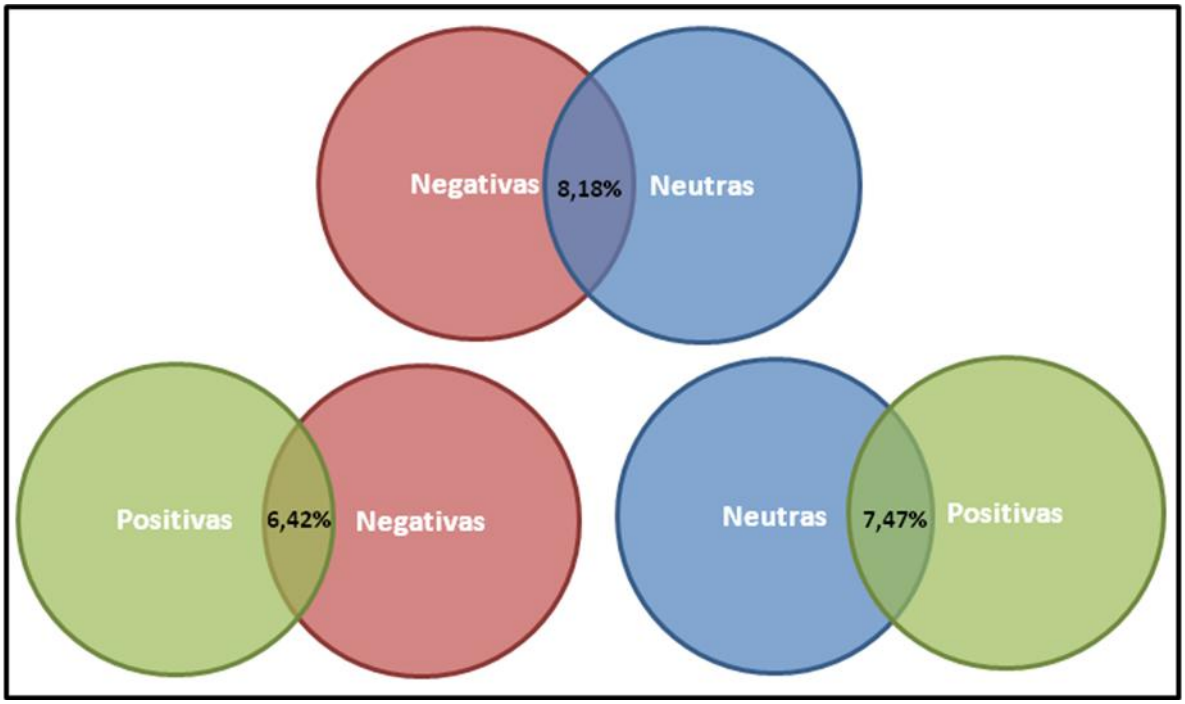


**Figura 4 - Diagrama de intersecção de ambigüidade.**

# 6 Conclusões e Considerações Finais

A proposta inicial do trabalho foi avaliar a possibilidade de se utilizar técnicas de *Data Mining* para extração automatizada do sentimento das mensagens de ambientes microblog. Para isso se fez necessário: coletar e armazenar mensagens oriundas do ambiente microblog *Twitter;* realizar análise manual das mensagens; realizar diferentes tipos de pré-processamento das mensagens, e; aplicar técnicas de validação do algoritmo.

Obtidos os resultados através de diferentes combinações de testes, chegou-se as seguintes conclusões:

- O algoritmo de aprendizado supervisionado de Bayes alcançou resultados com média de 80% de acerto na classificação das mensagens quando consideradas apenas as classes positiva e negativa;
- A presença da classe neutra no processo de classificação afeta o desempenho do algoritmo, pois gera ruído por aumentar a ambigüidade dos termos do conjunto de treinamento.
- Apesar de se acreditar que pré-processamentos como: remoção de *stopwords* e *stemming* fosse contribuir para o aumento do desempenho do algoritmo, percebe-se que o pré-processamento que apresentou o melhor desempenho foi o que considera apenas tokenização por bigrama.

Por fim, conclui-se que a utilização do algoritmo de aprendizagem supervisionado de Naive Bayes para classificação de mensagens postadas no ambiente de microblog *Twitter* alcança resultados satisfatórios considerando-se apenas as classes positivas e negativas na análise, e com pré-processamento de tokenização por bigrama nas mensagens.

# 7 Trabalhos Futuros

Esse trabalho mostrou os resultados alcançados na aplicação do algoritmo de aprendizagem supervisionado de Naive Bayes na classificação do sentimento de textos encontrados no *Twitter,* a partir de um conjunto de treinamento classificado manualmente. Para ampliar a aplicabilidade do que foi implementado algumas soluções podem ser aplicadas em trabalhos futuros:

- O conjunto de treinamento deve ser classificado de forma redundante sendo classificadas várias vezes por diferentes indivíduos. Isso ajudaria a corrigir problemas de subjetividade, tornando o conjunto de treinamento mais coerente.
- Aplicação de técnicas como fazer uma curva de aprendizado para avaliar qual seria a quantidade ideal de mensagens classificadas manualmente para a obtenção do conjunto de treinamento que alcance maior desempenho.
- Outro estudo incremental seria avaliar qual o correlação entre o que foi obtido pelo classificador, e o que foi observado na realidade, ou seja se é possível avaliar a existência de alguma relação na queda ou aumento nas vendas de um determinado produto e a opinião predominante sobre este nos micro blogs etc.
- Sabendo-se que uma quantidade considerável das mensagens postadas no *Twitter* possui links (para um vídeo ou texto de outro site), pode-se projetar *crawlers* para navegar até o site do link e obter os comentários registrados para o vídeo ou texto referido pelo link.

Por fim poderia ser usado um conjunto de classificadores para conseguir um classificador mais eficiente, e não apenas Naive Bayes como foi aqui implementado.

# 8 Referências

ALEC GO, RICHA BHAYANI AND LEI HUANG. **Twitter Sentiment Classification using Distant Supervision** CS224N Project Report, Stanford. 2009

BING, LIU. **Web Data Mining, Exploring Hyperlinks, Contents and Usage Data,** Springer, 2006.

CAMARGO YURI BARWICK LANNES: **De Abordagem Lingüística na Classificação Automática de Textos em Português**. Rio de Janeiro, COPPE/UFRJ, 2007. Acesso em 02 de mar de 2010. Disponível em: <http://www.pee.ufrj.br/teses/?Resumo=2007062502>

CHAKRABARTI, S. Mining the Web**: Discovering Knowledge from Hypertext Data**. San Francisco, USA: Morgan Kaufmann, 2003.

DUDA, R.O.; HART, P.E.; STORK, D.G., 2000, **Pattern Classification**, 2 Ed., California, Wiley Interscience.

FRAKES, W.B.; R.BAEZA-YATES. **Information Retrieval: data structures and algorithms**. London, UK: Prentice Hall, 1992.

JACKSON, P., MOULINIER, I., 2002, **Natural Language processing for Online Applications – Text retrieval, Extraction and Categorization**. Philadelphia: John Benjamins B.V.

KOSALA, R., BLOCKEET, H.: **Web Mining Research: A Survey. SIGKDD Explorations,** Vol. 2, Issue 1, July 2000.

LEWIS, D. D., 1992, **Representation and Learning in Information Retrieval**. PhD Thesis. Department of Computer and Information Science, University of Massachusetts, Massachusetts.

MANDEL, A.; SIMON, I.; DELYRA, J. L. **Informação: Computação e comunicação.** Universidade de São Paulo 1997. Acesso em: 20 nov de 2010. Disponível em : <http://www.ime.usp.br/ is/abc/abc/abc.html>.

MANNING C.D.; SCHUTZE H. **Foundations of Statistical Natural Language Processing**. MIT Press 1999

MITCHELL, T. M., 1997, **Machine Learning.** USA, Ed. McGraw-Hill.

P. Refaeilzadeh, L. Tang, and H. Liu. **"Cross Validation", in Encyclopedia of Database Systems (EDBS),** Editors: Ling Liu and M. Tamer Özsu. Springer, pp6. 2009

PANG, BO; LEE, LILIAN AND VAITHYANATHAN, SHIVAKUMAR. **Thumbs up? Sentiment Classification Using Machine Learning Techniques**, in Proceedings of EMNLP, 2002 pp 79-86

PIRES, MARINA MELO. **Agrupamento Incremental e Hierárquico de Documentos**, Rio de Janeiro, 2008. Dissertação apresentada para obtenção do grau de mestre ciências em Engenharia Civil - Universidade Federal do Rio de Janeiro , 2008.

SILVA, MAURO DA SILVA. **Uma Metodologia para Extração de Conhecimentos em Objetos Textuais Baseada em Conceito para o Português do Brasil**, 2007. Dissertação apresentada para obtenção do título de Mestre em Ciência da Computação – Instituto de Informática da Universidade Federal de Goias, 2007

*TWITTER*, WEB SITE OFICIAL. **About Twitter.** 2010: Acesso em: 20 mar de 2010. Disponível em:< http://*Twitter*.com/about >.

VENTURA, JOÃO MIGUEL JONES. **Extracção de Unigramas Relevantes**. Lisboa, 2008. Dissertação apresentada para obtenção do grau de Mestre em

Engenharia Informática - Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2008.