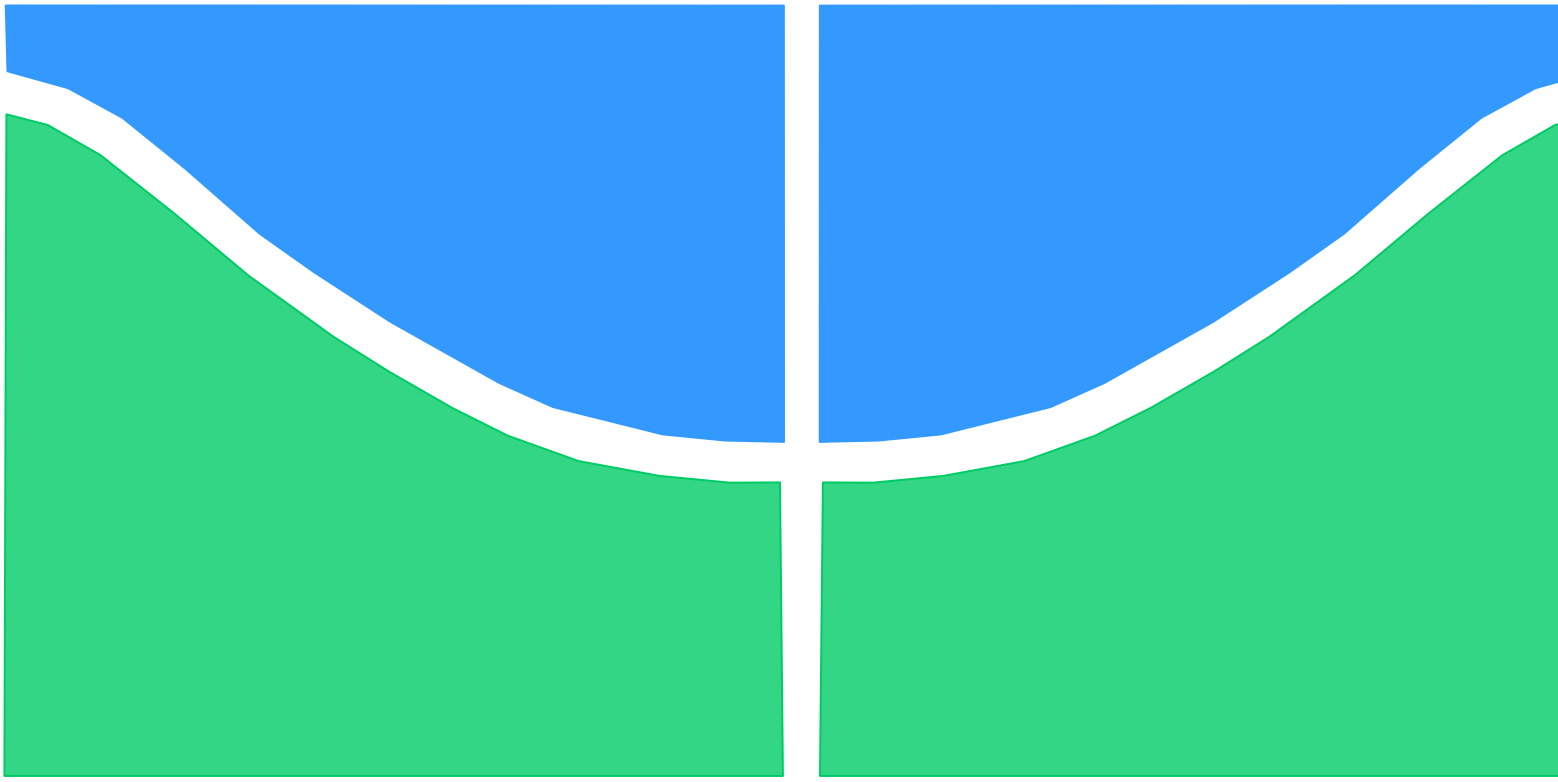

PROJETO DE GRADUAÇÃO

CLASSIFICADORES DE POLARIDADE DE NOTÍCIAS UTILIZANDO FERRAMENTAS DE *MACHINE LEARNING:* O CASO DA VALE S.A.

Por,
Filipe Guedes de Oliveira Almeida
09/0113756

Brasília, 09 de dezembro de 2014

UNIVERSIDADE DE BRASILIA

FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Departamento de Engenharia de Produção

PROJETO DE GRADUAÇÃO

CLASSIFICADORES DE POLARIDADE DE NOTÍCIAS UTILIZANDO FERRAMENTAS DE *MACHINE LEARNING:* O CASO DA VALE S.A.

POR,

Filipe Guedes de Oliveira Almeida

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Produção

Banca Examinadora

Profa. Dra. Ana Carla Bittencourt Reis, UnB/ EPR (Orientador) _____

Prof. Dr. Carlos Henrique Marques da Rocha, UnB/ EPR _____

Prof. Dr. João Carlos Felix Souza, UnB/ EPR _____

Brasília, 09 de dezembro de 2014

RESUMO

A dificuldade de se prever o movimento das ações é objeto de estudo de vários autores. A fim de obter ganhos imediatos, se faz necessário estimar a direção do movimento de curto-prazo para decisão do momento mais apropriado para negociar ações. A proposta desse trabalho consiste em selecionar a ferramenta de *machine learning* mais adequada para classificar a polaridade notícias da empresa Vale divulgadas para os investidores em geral. Também serão utilizadas ferramentas de *Natural Language Processing* (NLP) para pré-processar o texto e definir os parâmetros de pré-processamento que geram melhores resultados para o classificador. Para chegar a esta proposta, foi feita uma ampla revisão bibliográfica sobre NPL, *Machine Learning*, *text mining* e a influência de fatores macroeconômicos no valor das ações e vice versa. Dessa forma, foi possível selecionar as ferramentas mais adequadas para a realização da segunda etapa do projeto, que consistiu em gerar diversos classificadores e compará-los a fim de identificar os melhores parâmetros para pré-processamento, seleção de atributos e processamento das notícias.

Palavras-chave: *Machine Learning, Natural Language Processing, Naïve Bayes Multinomial, Support Vector Machine, Random Forest.*

ABSTRACT

The difficulty of predicting the stock prices movement is studied by several authors. In order to obtain immediate gains, it is necessary to predict the direction of the movement of short-term to decide the most appropriate time to trade the stocks. The purpose of this project is to select the most appropriate machine learning tool to classify the news polarity from VALE S.A. available online to the investors. The project used as well, natural language processing tools pro preprocess the text and define the best parameters to preprocessing. To achieve these proposal, an extensive literature review about NPL, Machine Learning, text mining and the macroeconomic factors that influence the stock prices and vice versa. Thus, it was possible to select the most appropriate tools to perform the project, which covered the generation of several classifiers and compare them to identify the best parameters of pre-processing, attribute selection and processing the news.

Keywords: *Machine Learning, Natural Language Processing, Naïve Bayes Multinomial, Support Vector Machine, Random Forest.*

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE FÓRMULAS

LISTA DE SÍMBOLOS

Siglas

ABNT	Associação Brasileira de Normas Técnicas
BOW	*Bag-of-words*
CFS	*Correlation Feature Selection*
EUA	Estados Unidos da América
FN	Falso negativo
FP	Falso positivo
IDF	*Inverse document frequency*
KNN	*K Nearest Neighbors*
LLSF	*Linear Least Square Fit*
MHS35	*Morgan Stanley High-Tech Index*
ML	*Machine Learning*
NASDAQ	*National Association of Securities Dealers Automated Quotations*
NBM	*Naive Bayes Multinomial*
NPL	*Natural Language Processing*
NYSE	*New York Stock Exchange*
PIB	Produto Interno Bruto
PLN	Processamento de Linguagem Natural
ROC	*Receiver Operating Characteristic*
S&P	*Standard & Poor's*
SELIC	Sistema Especial de Liquidação e de Custódia
SVM	*Support Vector Machine*
TF	*Term frequency*
VN	Verdadeiro negativo
VP	Verdadeiro positivo

1 INTRODUÇÃO

O valor das ações varia frequentemente e por vezes sem um padrão reconhecido. A fim de obter ganhos imediatos, se faz necessário prever a direção do movimento de curto-prazo para decisão do momento mais apropriado para negociar ações. Diversos estudos foram realizados para tentar buscar padrões de comportamento do mercado, porém, múltiplas variáveis influenciam a tomada de decisões dos investidores e são essas decisões que irão influenciar o valor da ação, o que dificulta a modelagem de uma função geral para prever o preço das ações. De acordo com Martinez *et al.* (2009), a dificuldade de se prever o valor das ações, se deve à natureza do mercado de ações, que pode ser caracterizado como: complexo, dinâmico, caótico e evolutivo. Segundo Mankiw (2010), testes estatísticos mostram que os preços das ações variam de maneira aleatória ou quase aleatória. Dessa forma, programar uma função que preveja o valor futuro demandaria grande trabalho.

Apesar de geralmente o processo de decisão de um investimento ser intuitivo, toda forma de investimento, requer uma análise prévia de avaliação do ativo buscando aumentar o retorno do investimento e, para tanto, diversas abordagens foram desenvolvidas para maximizar a lucratividade das ações. Esses modelos de avaliação, de acordo com Assaf Neto (2011), buscam projetar o comportamento futuro dos ativos financeiros. Cavalcante *et al.* (2009) afirmam que o sucesso do investimento em ações depende basicamente da capacidade de análise do investidor, o que sintetiza a importância das análises antes da decisão do investimento. As abordagens disponíveis utilizam análises que podem ser baseadas em fatores qualitativos ou quantitativos. Outra segmentação é entre análise gráfica ou fundamentalista. Na análise fundamentalista, onde são analisados os resultados setoriais e específicos de cada empresa dentro do contexto da economia nacional e internacional (Fortuna, 2011). De acordo com Pinheiro (2009), a análise fundamentalista consiste no estudo de toda informação disponível no mercado sobre determinada empresa, e com isso, o investidor decide pela compra ou venda da ação. Já a análise gráfica, de natureza mais quantitativa, trabalha com gráficos apoiados por métodos estatísticos. De acordo com Fortuna (2011), a escola gráfica tem como premissa de que uma boa análise gráfica dispensa a pesquisa dos fundamentos da empresa, tendo em vista que o gráfico sintetiza todos os conhecimentos, experiências e expectativas do mercado sobre aquela ação.

Diversas ferramentas buscando quantificar a análise qualitativa podem ser utilizadas. Mankiw (2010) define ações como cotas de propriedades de empresas, enquanto que o mercado de ações é o local onde ocorre a negociação das ações. O autor apresenta o

numerador q de Tobin, que pode ser interpretado como um indicador que reflete a lucratividade esperada do capital bem como a lucratividade corrente e pode ser representado pela Equação 1:

$$q = \frac{Valor\ de\ Mercado\ do\ Capital\ Instalado}{Custo\ de\ Reposição\ do\ Capital\ Instalado} \quad (1)$$

Lindenberg e Ross (1981) mostraram que o valor de mercado do capital instalado pode ser calculado como a soma do valor de mercado das ações com o valor de mercado das dívidas. O valor de mercado das dívidas deve ser definido com base no prazo de pagamento e taxa de juros. Já o valor de mercado das ações para empresas listadas em bolsa de valores é calculado pela multiplicação do número de ações listadas em bolsa multiplicado pelo preço de cada ação. Já o custo de reposição do capital instalado, ainda segundo Lindenberg e Ross (1981) refere-se ao valor de reposição dos ativos da empresa, ou seja, o valor necessário para renovar a capacidade produtiva atual. Caso o q de Tobin seja superior a 1, o mercado valoriza o estoque de capital mais do que o custo de reposição do capital. Nesse caso, tende-se a aumentar a demanda por capital, que, consequentemente gera um aumento do valor de mercado das ações desta empresa. Caso contrário, a tendência é de redução do valor das ações da empresa. (MANKIW, 2010).

Alguns economistas defendem a hipótese de mercados eficientes, onde de acordo com Pinheiro (2009) e com Mello e Spolador (2010), o preço das ações é reflexo de todas as informações relevantes disponíveis e o ajuste diante de uma nova informação é instantâneo, ou seja, os investidores monitoram os noticiários para definirem o momento de adquirir ou vender uma ação. Quando surgem boas notícias a respeito do futuro da empresa, o preço das ações cresce. Quando surgem notícias negativas, os preços caem. Essa hipótese trabalha, portanto, com a aleatoriedade do mercado. De acordo com essa teoria, o preço das ações não pode ser previsto com base nos preços anteriores, pois somente serão alterados à medida que novas ocorrências surjam e modifiquem as percepções dos investidores. Do outro lado, estão os economistas que rejeitam a hipótese de eficiência de mercado. Abdullah e Ganapathy (2000) defendem que é possível encontrar alguma previsibilidade com base nas séries temporais.

Blanchard (2011) mostra que o preço da ação é dado pelo valor presente do dividendo do próximo ano descontado a taxa de juros, conforme mostra a Equação 2 abaixo:

$$Q_t = \frac{D^e_{t+1}}{(1 + r_{1t})} + \frac{D^e_{t+2}}{(1 + r_{1t})(1 + r^e_{1t+1})} + \cdots \quad (2)$$

Onde Q_t é o valor da ação ao longo do tempo, D representa o dividendo pago aos acionistas por unidade de tempo e r a taxa de juros no momento t. Como mostra Pinheiro (2009), esse modelo parte da premissa de que a rentabilidade da ação é fornecida pelo dividendo distribuído pela empresa, porém, nem todas as empresas pagam dividendos a seus acionistas, sendo essa portanto, uma restrição desse modelo, como os casos de empresas em crescimento, que preferem reinvestir os lucros no negócio do que distribuir aos acionistas na forma de dividendos.

Analisando a equação, é possível concluir que mudanças na taxa de juros por parte do Banco Central, provocam variações do valor da ação. Reduções na taxa de juros provocam aumento do valor do ativo. Pela equação, entende-se também que o aumento do valor dos dividendos torna a ação mais atrativa, podendo provocar aumento do valor das ações. É possível deduzir então, que outro fator que pode influenciar o valor das ações são as políticas de governo. Qualquer decisão tomada pelo governo com relação a impostos de investimentos e taxas de juros pode migrar o tipo de investimento do investidor. Pode-se citar como exemplo, a redução da taxa SELIC em 2012 que chegou à 7,5% ao ano, em reportagem divulgada na seção de economia do portal de notícias IG. Essa queda reduziu a rentabilidade da poupança brasileira. Em um primeiro momento, essa decisão migrou parte do investimento que antes estava aplicado em poupança para investimentos de maior risco, porém com maior possibilidade de rendimento, como o mercado de ações, por exemplo, elevando o valor dos ativos.

A decisão de investir em ações de uma determinada empresa baseia-se na possibilidade de uma renda futura. Assim, de acordo com Mankiw (2010), os preços das ações tendem a subir quando as empresas possuem muitas oportunidades de investimento lucrativo. Portanto, o preço das ações é reflexo dos incentivos para se investir em um ativo.

Diversos autores como Mankiw (2010) e Mello e Spolador (2010), Mittermayer (2004), Hagenau *et al*. (2013) entre outros, referem-se que em grande parte dos casos, a expectativa do mercado é moldada com base nas notícias. Porém Blanchard (2011) mostrou que em alguns casos, uma notícia positiva pode gerar queda das ações, enquanto uma negativa pode resultar no aumento dos preços. Ele apresenta dois exemplos: um onde notícias positivas geraram redução no preço das ações e outro o impacto positivo causado por uma notícia ruim sobre a economia. O primeiro exemplo, de janeiro de 1999, "O anúncio dos números das vendas no varejo em novembro maiores do que o esperado não foi bem recebido. A economia

aquecida gera temores de inflação e aumenta o risco de que o Federal Reserve eleve as taxas de juros novamente." Blanchard (2011). O segundo exemplo, "Os investidores puseram de lado as notícias pessimistas sobre a economia e concentraram-se na esperança de que o pior já passou, tanto para a economia, quanto para a bolsa de valores.", divulgado em agosto de 2011 mostrou que apesar das notícias pessimistas sobre a economia, o índice da Nasdaq obteve aumento de 2%.

Huang *et al*. (2004) mostram que o valor individual de um ativo geralmente acompanha a tendência de todo o mercado de ações. O valor da ação da Vale, por exemplo, tende a subir quando o Índice Bovespa está em alta. Esse estudo defende que o movimento do mercado tem fortes influências sobre o preço do ativo.

Um fator que afeta principalmente o valor das ações em momentos de grande flutuação de preços diz respeito ao efeito manada (Yao *et al*., 2006). O efeito manada é explicado por Christie e Huang (1995) como o momento em que o investidor age de maneira irracional, deixando de lado suas análises e crenças e age somente baseado nas decisões coletivas do mercado, mesmo quando discorda dessas ações. O investidor age, portanto, de maneira impulsiva baseado no comportamento de outros investidores. Dessa forma, o comportamento de manada pode ter impacto extremamente negativo no valor das ações em momentos de declínio de mercado. Porém, autores como Bikhchandani e Sharma. (2000), buscam mudar a visão pejorativa do efeito manada. O estudo desenvolvido defende que esse tipo de atitude pode ser realizado de maneira deliberada. Os autores argumentam que os demais investidores sabem algo a respeito do retorno do investimento e suas ações refletem isso. Ou seja, essa teoria defende que o efeito manada pode ser algo intencional de copiar o comportamento de outros investidores.

Corroborando com as pesquisas citadas anteriormente, Chang *et al.* (2000) em estudo feito nos mercados de ações dos EUA, Hong Kong, Japão, Coréia do Sul e Taiwan, mostram que geralmente, informações macroeconômicas tendem a influenciar mais o comportamento do investidor do que as próprias informações específicas da empresa.

Apesar de inúmeros fatores moldarem o valor das ações, situações inversas também podem ser percebidas, onde a variação do preço da bolsa de valores influência a economia. Ao contrário do exposto na seção anterior, onde geralmente os fatores macroeconômicos impactavam no movimento do preço das ações, existem casos que uma variação no mercado financeiro afetou a economia. Um exemplo claro foi o forte declínio da bolsa de valores norte-americana, que foi um dos fatores que determinou a recessão em 2001, como mostra Blanchard (2011). O mesmo autor apresenta outros exemplos como a quebra da bolsa em

1920 que desencadeou na Grande Depressão e o declínio do índice Nikkei como uma das causas da crise japonesa da década de 90.

A Figura 1 mostra que em alguns casos, a variação do mercado de ações reflete variações no PIB (Produto Interno Bruto) real. Um exemplo claro diz respeito à crise de 2008. Conforme pode ser observado pela Figura 1, períodos anteriores a grande queda da curva do PIB real, houve uma queda acentuada dos preços das ações.

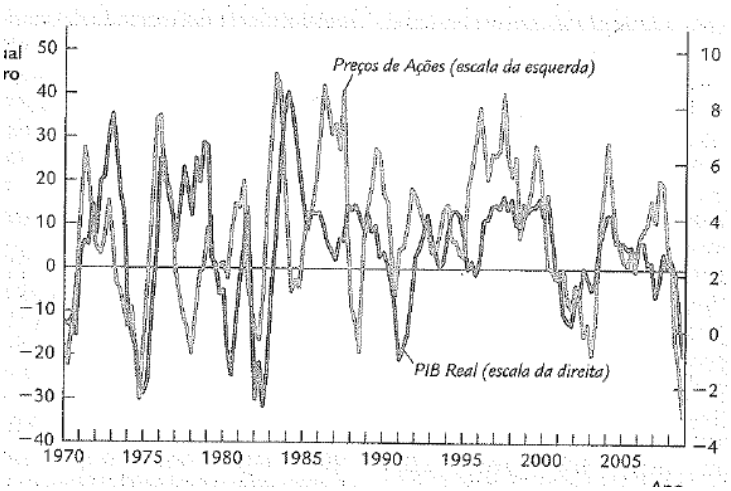

Figura 1 - Evolução do Preço das Ações x PIB Real
Mankiw (2010)

Conforme explicitado por Mankiw (2010), grandes quedas das bolsas de valores, pode significar que uma recessão está se aproximando. A variação no valor das ações pode também estar associada à atividade econômica. Mankiw (2010) argumenta que as ações representam parte do patrimônio das famílias, e uma vez que haja queda no valor das ações, parte desse patrimônio é perdido, pressionando para baixo os gastos e a demanda. O contrário também é verdadeiro. O aumento do valor das ações pode representar aquecimento da atividade econômica, com o aumento da compra.

Outra forma de identificar o momento de compra e venda de ações é realizada de maneira quantitativa, baseada principalmente na análise gráfica. Como mostra Pinheiro (2009), a hipótese da análise gráfica é de que é possível realizar previsão do movimento futuro das ações com base nos movimentos históricos. Porém, de acordo com Tan *et al*. (2005), esse tipo de análise é mais propensa a erros devido justamente às características do mercado expostas anteriormente, como sua complexidade e dinamicidade. Apesar disso, a ferramenta estatística de análise gráfica ainda é utilizada por diversos analistas financeiros. As abordagens qualitativas surgiram para contrapor a corrente apresentada anteriormente e leva em consideração fatores como decisões políticas macroeconômicas, políticas da empresa,

condições econômicas gerais (Oh e Kim, 2002), psicologia dos investidores (Kuo *et al*., 2001), expectativas dos investidores, variação de preço de outros ativos (Huang *et al*., 2004), etc. para basear a análise de comprar ou vender a ação. Com o avanço das tecnologias de informação e agilidade de processamento, nos últimos anos, surgiram diversas abordagens utilizando ferramentas e técnicas de inteligência artificial em busca de maior rentabilidade nos ativos (Gomide e Milidiú, 2010). Essas técnicas buscam não só entender padrões na variação dos preços, mas também o comportamento do investidor na hora de tomar decisões, utilizando tanto fatores quantitativos quanto qualitativos.

A escolha do investidor por comprar uma determinada ação, é geralmente baseada em uma expectativa de que no futuro esse ativo gere retorno. Quanto maior for a expectativa de mercado, maior a tendência de alta do preço. Essa expectativa pode ser de um possível dividendo que será pago aos acionistas ou uma possibilidade de vender a ação por um valor superior ao valor investido. Posto isso, uma abordagem mais recente utiliza notícias disponíveis em jornais a respeito da empresa ou do setor para identificar correlações entre a notícia e o valor das ações. As notícias trazem informações qualitativas a respeito das ações da empresa e do setor que influenciam a expectativa dos investidores e consequentemente, da compra e venda de ações. Contudo, com o aumento significativo das informações disponíveis e a velocidade com que as informações são geradas se torna difícil para o investidor o acompanhamento em tempo real de todas as notícias disponíveis. Dessa maneira, o estudo desenvolveu uma forma automatizada de classificar as notícias de acordo com a sua polaridade utilizando ferramentas de *Machine Learning* e *Natural Language Processsing* (NLP). Foi utilizado conhecimento de NLP para transformar a base de notícias inicialmente desestruturada em formato de texto, em uma base estruturada e possível de ser analisada de maneira mais simples. Estudos mostram que classificar as notícias pode fornecer informações adicionais que podem ser usadas para prever tendências de preços das ações (Mittermayer, 2004). Em seguida, foram aplicados diversos algoritmos de *Machine Learning* para criação de classificadores e entendimento de padrões de influência das notícias no preço das ações. Gidofalvi (2001) mostra que é possível prever o movimento de curto prazo do valor das ações baseado em notícias. Foram então criados vários classificadores de notícias e comparados para seleção daquele que obteve melhor desempenho. Dessa forma, partindo da premissa de que as notícias da empresa e do setor influenciam o valor das ações de curto prazo, o trabalho selecionou o melhor classificador de notícias para previsão da direção do movimento dos preços das ações da Vale utilizando ferramentas de *Machine Learning*. Cabe ressaltar, que os

algoritmos de NLP são utilizando somente para facilitar a criação dos classificadores de Machine Learning sendo, portanto, uma etapa preliminar aos classificadores.

Esse relatório está dividido da seguinte forma: o capítulo 2 aborda uma revisão de literatura sobre conceitos e ferramentas de *machine learning*, *natural language processing e text mining*. No capítulo 3, é apresentada uma visão geral dos trabalhos relacionados a este, seja por utilizar ferramentas de *text mining* ou *machine learning* associados ao mercado de ações. No capítulo 4 será apresentada a metodologia de trabalho para atingir os objetivos propostos a seguir, seguido do capítulo 5 com o desenvolvimento das etapas do trabalho e resultados alcançados. O capitulo final compreenderá a conclusão do trabalho.

1.1 OBJETIVO GERAL

O objetivo do trabalho é selecionar os melhores parâmetros de pré-processamento para selecionar o classificador que gera melhores resultados na classificação da polaridade das notícias da Vale S.A.

1.2 OBJETIVOS ESPECÍFICOS

- Analisar diferentes ferramentas de *machine learning*;
- Identificar parâmetros de pré-processamento que mais influenciam no resultado dos classificadores de notícias para o caso em estudo;
- Identificar palavras ou conjunto de palavras que possuem tendências positivas e negativas para o caso em estudo;
- Identificar palavras ou conjunto de palavras que geram maior impacto para classificação da notícia em positiva ou negativa para o caso em estudo;
- Classificar notícias com boa acurácia utilizando ferramentas de *natural language processing e machine learning*.

2 BASE CONCEITUAL

A base conceitual apresentada a seguir contemplou uma extensa revisão bibliográfica buscando entender as ferramentas de inteligência artificial que auxiliam no processo de mineração de textos. Dentro das ferramentas de mineração de texto, foi abordado especificamente sobre *natural language processing*, que trabalha com o pré-processamento do texto e o transforma em uma linguagem mais estruturada e simples de ser analisada e *machine learning*, que extrai informações do banco de dados e aprende essas informações para criar um classificador para classificar novas informações inseridas dentro do banco de dados.

2.1 TEXT MINING OU MINERAÇÃO DE TEXTO

Os métodos de *text mining* ou mineração de texto tem como objetivo a descoberta de padrões em uma coleção de dados não estruturados. Ou seja, é um processo de extração de conhecimento a partir dos textos sem a necessidade de lê-los por completo. É nessa etapa que será realizada a representação do conhecimento em um ormato possível de ser lido pelo computador (Hagenau *et al*., 2013). Cabe ressaltar que a mineração de texto é diferente da busca por palavras chaves. Consiste em uma ferramenta muito mais sofisticada e analítica. Enquanto na busca já é sabido o que se quer buscar, em *text mining* descobre dados inicialmente desconhecidos. Para realizarmos a descoberta dos dados, é possível utilizar algoritmos de *machine learning e Natural Language Processing* que auxiliam na modelagem e estruturação da base textual.

2.2 *NATURAL LANGUAGE PROCESSING* OU PROCESSAMENTO DE LINGUAGEM NATURAL

O processamento de linguagem natural é uma área da inteligência artificial capaz de converter uma informação, como um texto, por exemplo, em algo mais fácil de ser analisado por um computador ou uma pessoa. Essa etapa é responsável por facilitar o processo seguinte, por meio de um pré-processamento para realizar essa conversão e reduzir o texto selecionando os recursos mais adequados. Por recursos adequados, deve-se entender como palavras mais

apropriadas para realizar a análise. A seleção dos recursos, de acordo com Forman (2003), é necessária para tornar problemas grandes mais eficientes computacionalmente. Além disso, Forman (2003) defende que a seleção correta dos recursos pode melhorar consideravelmente a acurácia da classificação, necessitando de menor quantidade de dados para teste para chegar ao nível de performance desejado. Para Oguri *et al*. (2006), o primeiro passo para a classificação do texto é transformar documentos numa representação adequada para o algoritmo de aprendizado e isso é conseguido por meio de ferramentas de Processamento de Linguagem Natural (PLN). O PLN estuda a maneira como os computadores podem ser utilizados para compreender e manipular a linguagem natural (Chowdhury, 2003 *op. cit.* Gomes, 2013) e com isso possui papel essencial para a mineração de texto, pois realiza um pré-processamento do arquivo que possibilita uma primeira estruturação dos dados. O PLN é uma área muito ampla com diversas ferramentas. Gomes (2013) a partir do estudo de Feldman (1999) mostra a diferenciação dos vários níveis que o PLN pode trabalhar. São segundo ele, seis níveis: morfológico, léxico, sintático, fonético, semântico e pragmático. O primeiro trabalha com a estrutura e forma das palavras e o léxico, com o significado das palavras. O sintático estuda a gramática, o fonético a pronúncia, o semântico trabalha com a tradução do significado das palavras e frases e o pragmático com o conhecimento das pessoas.

Por conta da grande amplitude dessa área do conhecimento, serão abordadas nessa etapa, somente as ferramentas que serão utilizadas para pré-processamento, mais especificamente relacionadas ao morfológico, como *bag-of-words*, *n-gram* e lista de *stopwords.*

2.2.1 *TOKENIZATION* OU SEGMENTAÇÃO DE PALAVRAS

Tokenization é o processo que segmenta o texto em palavras, onde cada palavra será um *token*, que é uma divisão feita com base nos espaços entre uma palavra e outra. Utilizando a oração “O Brasil vai ser hexa.”, o processo dividirá em cinco *tokens*. [O] será o primeiro *token*, [Brasil] o segundo, [vai] será o terceiro *token*, [ser] o quarto e [hexa] o quinto. Com isso, obtém-se um conjunto de dados mais fáceis de serem analisados do que o texto completo. Porém, a divisão dos *tokens*, conforme alertado por Gomes (2013) gera inúmeras dimensões a serem analisadas, por isso, algumas palavras serão retiradas no processo de eliminação de *stop words*, que será explicado no item 2.2.4.

A segmentação de palavras é realizada principalmente de duas maneiras, utilizando *bag-of-words* ou *n-gram*.

2.2.1.1 *BAG-OF-WORDS* OU SACO DE PALAVRAS

Na abordagem *bag-of-words*, cada texto é representado como um vetor de palavras que ocorrem no documento. Se a palavra aparece no documento, recebe o valor 1 e caso contrário, valor 0. Dessa forma, é feita uma matriz semelhante à apresentada na Figura 2, onde w_i representa uma palavra e d_i representa um documento.

	w_1	w_2	w_3	...	w_n
d_1	1	0	0	...	1
d_2	0	0	1	...	1
d_3	1	1	0	...	0
⋮			⋱		⋮
d_m	0	0	1	...	0

Figura 2 - Exemplo de *bag-of-words*
Elaborado peloguri *et al*. (2006)

Oguri *et al*. (2006) ressaltam que nessa abordagem a ordem com que as palavras ocorrem é ignorada. Hagenau *et al*. (2013) mostram que pelo fato do *bag-of-words* realizar a separação somente em grupos de uma palavra, o sentido semântico das palavras não é capturado.

2.2.1.2 *N-GRAM* OU N-GRAMA

Diferente do *bag-of-words*, o n-grama busca resolver o problema da ordem das palavras e o sentido semântico, representando um conjunto de N palavras consecutivas. Um exemplo apresentado por Oguri *et al*. (2006) e que mostra a vantagem dessa abordagem é para o caso das expressões “homem grande” e “grande homem”, que no *bag-of-words* possuem a mesma classificação. Esse exemplo mostra que mesmo um modelo 2-grama já pode gerar ganhos se comparado com o anterior. Por outro lado, essa abordagem torna-se mais complexa e difícil de ser desenvolvida.

2.2.2 PART-OF-SPEECH TAGGING

Definidos os *tokens*, o processo de *part-of-speech tagging*, define uma marca para cada *token* a partir de uma base gramatical ou dicionário, por exemplo. Essa etapa consegue extrair para cada *token* a sua classificação gramatical, ou seja, se as palavras são substantivos,

adjetivos, verbos, advérbios, pronomes, etc. Mantendo o exemplo anterior, [O] seria classificado como artigo, [Brasil] como substantivo, [vai] e [ser] verbo e [hexa] adjetivo.

2.2.3 *NAMED ENTITIES* OU ENTIDADES NOMEADAS

O processo de identificação de entidades nomeadas é basicamente identificar nomes próprios como nomes de pessoas, organizações, locais, etc. No exemplo "O Brasil vai ser hexa.", o *token* [Brasil] seria identificado como uma entidade nomeada.

2.2.4 ELIMINAÇÃO DE *STOP WORDS*

A última etapa de pré-processamento é a eliminação de *stop words*, que são geralmente palavras sem valor semântico ou que fornecem pouca informação. Esse processo é realizado, conforme já apresentado anteriormente, para reduzir a dimensão da análise de modo a ser feita mais rápida. Geralmente são excluídos artigos e preposições e dependendo do método, com base na ocorrência. Quando trabalha-se com base na ocorrência, assume-se que as palavras que mais se repetem trazem maiores informações, enquanto a frequência inversa assume que as palavras mais raras têm maior poder de informação. Mantendo o exemplo "O Brasil vai ser hexa", no processo de eliminação de *stop words*, como estamos trabalhando com um exemplo pequeno onde não há repetição de palavras, seria retirado somente o artigo do primeiro *token* [O].

2.2.5 SELEÇÃO DE ATRIBUTOS

Em classificadores textuais, na maioria das vezes, e etapa de *tokenization* gera muitos atributos, o que pode dificultar o processo de classificação causado por atributos irrelevantes. Quanto maior a presença de atributos irrelevantes e redundantes, maior a dificuldade do aprendizado do classificador durante a etapa de treino. Uma maneira de retirar os atributos irrelevantes é selecionando os atributos mais importantes para a classificação, ou seja, aqueles que possuem maior poder de diferenciar entre as notícias positivas e negativas. Portanto, algoritmos de seleção de atributos são utilizados para remover atributos redundantes e irrelevantes (Liu, 2011). Witten *et al*. (2011) mostram que apesar de inicialmente se pensar na relação lógica de que quanto mais atributos, maior o poder de descrição, na prática, a presença de atributos irrelevantes em uma base, geralmente confunde o classificador. É possível concluir então, que a seleção de atributos é o processo de identificação e remoção do máximo de atributos irrelevantes possíveis e com isso, permite que os algoritmos aprendam de maneira mais rápida e eficiente.

Quando falamos de modelos baseados em árvores de decisão, a remoção de atributos pode produzir árvores menores e mais eficientes. Nos modelos bayesianos, que adotam a premissa de que os atributos são independentes, ficam mais claros os benefícios que a remoção dos atributos redundantes pode gerar (Langley e Sage, 1994). Os modelos baseados em instâncias, como o KNN, possuem maior probabilidade de sofrer grande influência dos atributos irrelevantes, à medida que seleciona as instâncias mais próximas para gerar a classificação, ou seja, poucas instâncias são consideradas para gerar a decisão. Portanto, a hipótese inicial é que os algoritmos de seleção de atributos irão tornar o classificador mais simples e com melhor acurácia.

Witten *et al*. (2011) concluem que não existe uma medida universal de relevância de atributos, portanto alguns serão testados nesse projeto a fim de verificar qual possui melhor ganho para a situação analisada. Witten *et al*. (2011) mostram também que além dos métodos de seleção mencionados, é possível fazer a seleção dos atributos baseado em um algoritmo de *machine learning*. São apresentados exemplos de seleção de atributos utilizando algoritmos de árvore de decisões para selecionar os principais atributos utilizados na árvore e modelos lineares, como SVM e os atributos são baseados nos coeficientes e remove os menores. A seleção também pode ser feita utilizando métodos baseados em instâncias. São checados os atributos próximos presentes na mesma classe e em classes diferentes para realizar a seleção. Então, esse tipo de seleção de atributos utiliza um algoritmo de aprendizagem para selecionar os atributos para outro. Porém, uma desvantagem apresentada por Witten *et al*. (2011) é que esses métodos não conseguem perceber a redundância de um atributo causada pela correlação com outro.

O algoritmo de seleção de atributos *correlation feature selection* (CFS) é baseado em correlações. O CSF baseia-se na correlação entre o atributo e a classe. Se a correlação entre a classe e o atributo é alta e a correlação entre o atributo e a outra classe é baixa, então o atributo é relevante. E em casos de correlação fraca ou forte para ambas as classes, o atributo é irrelevante. Outro benefício desse método é que ele também calcula o coeficiente de correlação entre os atributos buscando remover atributos redundantes.

Outro algoritmo para seleção de atributos que apresenta bons resultados segundo Feldman e Sanger (2007) é o qui-quadrado, que calcula a relação de dependência entre a classe e o atributo utilizando a estatística qui-quadrado. Hagenau *et al*. (2013) explicam que o qui-quadrado compara a frequência observada *Oi* do atributo *i* dentro do conjunto de mensagens positivas com a frequência esperada e normaliza o desvio quadrado. Esse processo é repetido para os casos negativos. A soma dos dois desvios normalizados constitui a

estatística qui-quadrado. O grupo de palavras que possui maior valor, possui maior influência na classificação das notícias. Nesse método de seleção de atributos, diferente do CSF, quem define a quantidade de atributos que irão permanecer é o usuário.

Dentre as ferramentas mencionadas nessa seção, foram utilizadas *tokenization*, lista de *stop words*, e seleção de atributos. A justificativa para a escolha de cada uma dessas ferramentas no trabalho será explicada com detalhes na seção 4.

2.3 MACHINE LEARNING OU APRENDIZADO DE MÁQUINA

Machine Learning é um campo dentro da Inteligência Computacional que a partir de uma amostra de dados, estuda o desenvolvimento de métodos capazes de extrair conceitos (Mitchell, 1997). De acordo com Marsland (2009), algoritmos de *machine learning* são capazes de fazer com que o computador trabalhe para modificar ou adaptar suas ações de modo a obter melhores resultados. Ou seja, o computador aprende a partir de uma base de dados para melhorar sua acurácia e consequentemente, seu desempenho. Portanto, a coleta dos dados e como processar esses dados em informações, se torna crítico para se chegar a uma previsão com alto grau de acerto. Marsland (2009) define *machine learning* (ML) como algo que se torna melhor em alguma tarefa por meio da prática, que em termos mais específicos, quer dizer que o algoritmo aprende para identificar padrões. A principal vantagem do ML com relação às outras abordagens disponíveis, é que ele trabalha a partir de dados, e não de hipóteses.

Os algoritmos de *machine learning* podem ser classificados em quatro grandes grupos: aprendizado supervisionado, não supervisionado, reforçado e evolucionário. O primeiro e mais comum entre eles, consiste em fornecer um conjunto de dados com as respostas corretas como exemplo a partir do qual o algoritmo conseguirá generalizar e fornecer a resposta correta para as entradas futuras (Marsland, 2009), ou seja, o algoritmo é treinado para aprender uma função desejada. Essa base de treino possui como objetivo validar a eficiência do algoritmo. Na aprendizagem não supervisionada, não são fornecidas as respostas corretas. O algoritmo identifica similaridades entre os dados e os agrupa. A aprendizagem reforçada, de acordo com Marsland (2009) é um misto das duas anteriores. É informado para o algoritmo quando ele está errado, mas não se informa como corrigir. O algoritmo evolucionário consiste em melhorar sua previsão ao longo do tempo.

A principal utilização do *machine learning* neste trabalho será por meio dos classificadores. O principal objetivo do classificador é associar objetos de classes desconhecidas a um conjunto pré-definido de classes. Os classificadores devem ser capazes de prever em qual classe a notícia deve se enquadrar a partir de um treinamento (aprendizagem supervisionada). Dentre classificadores que trabalham com aprendizagem supervisionada, os principais são os modelos lineares, modelos baseados em árvores de decisão, modelos bayesianos e modelos baseados em entidades.

2.3.1 MODELOS LINEARES

Os modelos lineares, geralmente baseados em uma regressão linear podem ser utilizados para problemas de classificação binária. Conforme apresenta Witten *et al*. (2011) o resultado da regressão é uma linha que divide as duas classes. Para casos de inúmeros atributos, a linha de decisão entre as classes será um plano com k dimensões, onde k será o número de atributos. A regressão que define a classe é definida por:

$$x = w_o + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k \quad (3)$$

Onde x é a classe, w_o, w_1, w_2,..., w_n são os pesos de cada atributo e a_1, $a_2, \ldots, a_k$ são os valores dos atributos. Os pesos de atributo são calculados a partir da base de treino e que auxiliarão na definição da classe. É possível perceber que caso a relação entre os dados não seja linear, esse tipo de modelo pode não se enquadrar muito bem.

O modelo linear mais utilizado para classificação textual e que apresenta melhores resultados é o Support Vector Machine (SVM). Witten *et al*. (2011) explicam que o SVM seleciona um pequeno número de fronteiras críticas para cada classe chamadas de *support vectors* e define uma função linear que as divide.

O SVM foi introduzido por Vapnik em 1992 e vem sendo bastante utilizado graças aos melhores resultados que oferece se comparado com outros classificadores (Marsland, 2009). A principal característica do SVM é sua boa capacidade de generalização. A classificação é feita a partir dos dados de treino e da aprendizagem supervisionada, onde ele busca em sua memória informações parecidas e generaliza para se obter uma conclusão, mesmo de dados não presentes durante o treinamento. Quanto melhor a capacidade de generalização do classificador, melhor será sua acurácia. No SVM, inicialmente serão definidos dois conjuntos dentro de um espaço vetorial de duas dimensões. Acima da divisão, recebe classificação positiva, abaixo dele, classificação negativa. Essa divisão entre os dois conjuntos, geralmente é feita linearmente e está representada na Figura 3.

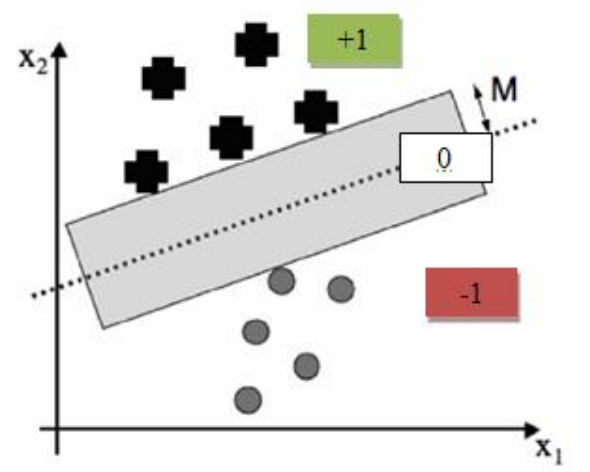

Figura 3 - Exemplo classificação SVM
Fonte: Adaptado de Marsland (2009)

Porém, nem todas as situações as fronteiras serão linearmente separáveis. Podem existir casos onde a fronteira é curva. Nesses casos, como mostra a Figura 4, realizar uma separação linear não retrata a realidade dos dados. O SVM resolve esse problema ao incluir também, termos não lineares na função e então, utiliza modelos lineares para implementar as fronteiras não lineares.

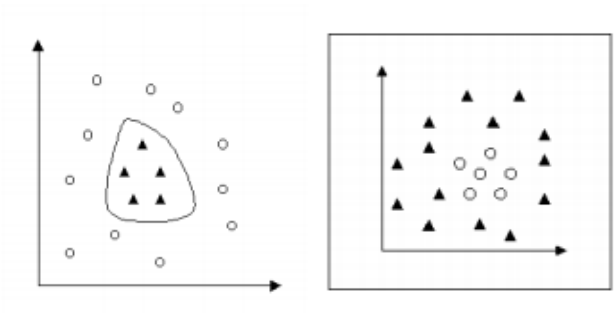

Figura 4 - Exemplo de conjunto linearmente separável (3a) e conjunto não linearmente separável(3b)
Fonte: Marsland (2009)

2.3.2 MODELOS DE ÁRVORES DE DECISÃO

Os modelos de árvore de decisão trabalham de cima para baixo. Cada nó de árvore especifica um teste de algum atributo e cada ramo de um nó representa um dos valores possíveis para o atributo testado. O resultado do classificador corresponde às folhas da árvore. O primeiro atributo a ser classificado e que é definido como a raiz, é o que melhor consegue representar uma classificação a partir dos dados de treino, ou seja, o que melhor separa os exemplos do treino. Para cada possível valor desse atributo é gerado um ramo e em seguida

selecionado o próximo atributo. E assim sucessivamente. Quanto mais difícil definir os limites entre cada grupo de possível classificação, mais ramos e nós a árvore terá e consequentemente, mais complexo será o classificador. De maneira simples, o classificador baseado em árvore de decisão pode ser exemplificado como na Figura 5.

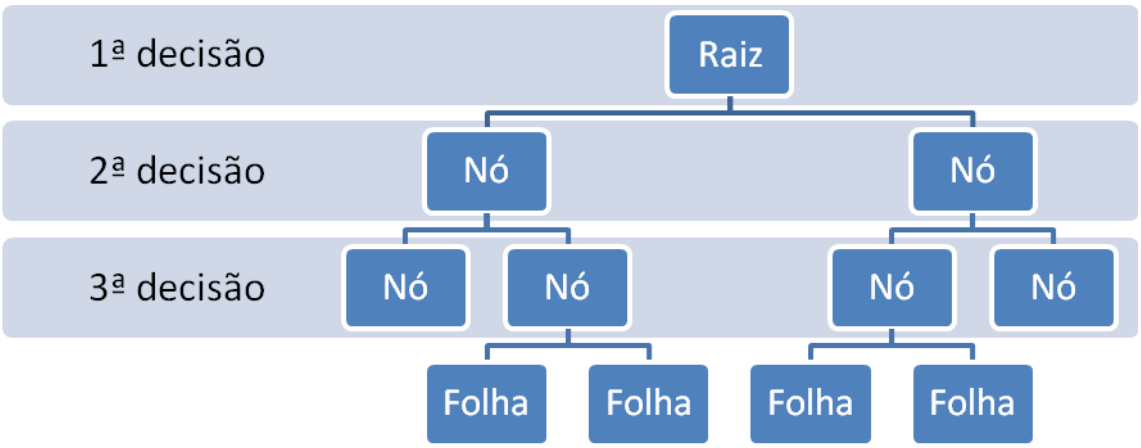

Figura 5 - Exemplo de árvore de decisões
Fonte: Elaborado pelo autor

Como pode ser visto na Figura 5, uma vantagem dos classificadores baseados em árvore de decisões é a sua facilidade de ser entendido (Feldman e Sanger, 2007). De acordo com Witten *et al*. (2011) o modelo de árvore de decisões mais utilizado atualmente é o C4.5. Essa abordagem trabalha com a construção da árvore de decisões a partir dos dados de treinamento e em seguida converte em um conjunto de regras, onde para cada caminho da raiz até a folha existe uma regra. E em seguida "poda" as regras pouco eficientes (grande taxa de erros) visando melhorar o desempenho do classificador. Por fim, ordena as regras de acordo com seu desempenho estimado e ao rodar o classificador, testa inicialmente as regras com melhores resultados para classificar as novas instâncias. O classificador C4.5 é um aprimoramento do ID3, que cria a árvore de decisão e o C4.5 realiza tratamentos adicionais à árvore, como a poda da árvore e criação de regras.

Outro método bastante utilizado para classificações textuais é o *Random Forest*. Diferentemente do C4.5, ele divide a base de dados de treino em várias subamostras aleatórias (*bagging*) e constrói uma árvore para cada subamostra (*boosting*). Para realizar a classificação de uma nova amostra, são utilizadas as árvores de todas as amostras. Marsland (2009) explica

que o método *boosting* parte da premissa de que uma série de classificadores fracos, quando colocados juntos, criam um conjunto de algoritmos de aprendizagem que podem gerar melhores resultados. É possível perceber que os métodos baseados em árvores de decisões são mais simples que os demais. Feldman e Sanger (2007) afirmam que a performance dos modelos baseados em árvores de decisão são inferiores que a maioria dos classificadores. Porém, mesmo assim, existem casos onde algoritmos mais simples conseguem obter melhores resultados que algoritmos robustos, e portanto, devem ser empregados para fins de comparação com outros classificadores.

2.3.3 MODELOS BAYESIANOS

Os modelos bayesianos, baseados no teorema de Bayes trabalham com distribuições de probabilidade (Witten et al. 2011), ou seja, a probabilidade de uma notícia pertencer a cada classe. A cada exemplo da base de treino pode alterar a probabilidade de cada classe da variável. Dessa forma, o algoritmo bayesiano calcula a probabilidade de cada classe ocorrer dados os atributos presentes e retorna a mais provável. As probabilidades são calculadas com base nos dados de treino.

A probabilidade de um atributo (xt) pertencer a classe (cj) de acordo com um modelo bayesiano é dada por:

$$P(cj|xt) = \frac{P(xt|cj)\,P(cj)}{P(xt)} \quad (4)$$

Onde a probabilidade da classe j dado que o vetor x_t está presente é calculada com base na probabilidade de ocorrer a classe c_j, a probabilidade de ocorrer o vetor x_t e a probabilidade de ocorrer o vetor x_t dado que ele está na classe c_j. Portanto, o classificador calcula a probabilidade de cada uma das classes e retorna a classe que maximiza a probabilidade.

Baseado nos estudos de Li (2009) é possível perceber que a principal diferença com relação aos classificadores anteriores é a capacidade de gerar estimativas de probabilidade e não somente a classificação. Enquanto os demais trabalham com um resultado exato, modelos bayesianos trabalham com uma probabilidade de uma alternativa ser verdadeira. Sabendo isso, é possível caracterizar os classificadores bayesianos como modelos probabilísticos.

O modelo bayesiano mais utilizado é o *Naive Bayes*. De acordo com Witten *et al*. (2011), o método deriva das regras de Bayes e faz uma simplificação assumindo que os

atributos são independentes, ou seja, adota-se a premissa de que a presença de uma palavra no texto não é afetada pela presença da outra, ou seja, as palavras são independentes. Essa premissa é adotada para fins de simplificação do algoritmo, porém é sabido que ela não é verdadeira, mas mesmo assim resultados empíricos mostram que isso pouco influencia no resultado final (Li, 2009). Langley *et al*. (1992) mostraram também que o classificador *Naive Bayes* apresentou performance surpreendentemente boa.

A probabilidade de uma notícia qualquer (n_j) ser classificada na classe (c_j) é igual à probabilidade combinada de todos os atributos presentes na notícia (n_j) pertencerem à classe (c_j). A probabilidade combinada de todos os atributos pertencerem à c_j é igual ao produto das probabilidades de cada atributo presente na notícia estar presente na classe (c_j) multiplicado pela probabilidade de ocorrer a classe (c_j), conforme apresenta a Equação 5:

$$P\big((cj|nj)\big) = P(cj) * \prod_i P(ai|cj) \tag{5}$$

A função que obtiver maior valor será a classificação mais provável para o vetor analisado.

Para a situação que será analisada mais a frente, como teremos somente a classificação positiva ou negativa, serão calculadas duas probabilidades:

$$\mathrm{P}(positivo|nt) = P(positivo) * \prod_i P(ai|positivo) \tag{6}$$

$$\mathrm{P}(negativo|nt) = P(negativo) * \prod_i P(ai|negativo) \tag{7}$$

Caso $\mathrm{P}(positivo|nt)$ seja maior que $\mathrm{P}(negativo|nt)$, a notícia é classificada como positiva. Caso contrário, negativa. É possível perceber com as equações acima, que o modelo *Nayve Bayes* assume que os valores de cada atributo são condicionalmente independentes.

Uma das falhas do *Naive Bayes* é que se um determinado atributo não aparece em uma determinada classe na base de treino, a probabilidade daquele atributo será 0 e como a probabilidade da classe é calculada pela multiplicação da probabilidade de todos os atributos, logo a probabilidade da classe será 0. Outra limitação do modelo *Naive Bayes* como apresenta Witten *et al*. (2011), é que para o caso de classificações utilizando textos, ele não leva em

consideração a quantidade de vezes que as palavras aparecem, e essa informação pode ser importante para a classificação. Para corrigir o primeiro problema, é possível incluir um fator de correção da equação que retire essa limitação. Para o segundo caso, foi criado o modelo *Naive Bayes Multinomial*, que é calculado de acordo com a Equação 8:

$$P\big((nj|cj)\big) = N! * \prod_{i=1}^{k} \frac{Pi^{ni}}{ni!} \tag{8}$$

Onde, N representa o numero de palavras no documento, Pi é calculada com base na frequência da palavra i na base de treino pertencendo à categoria nj e ni representa o número de vezes que a palavra i aparece no documento.

2.3.4 MODELOS BASEADO EM INSTÂNCIAS

Os modelos baseados em instâncias, como o nome sugere, realiza a classificação com base nas instâncias, que são os exemplos da base de treino. Dessa maneira, cada nova instância é classificada de acordo com os dados anteriores. É possível concluir que não há um "modelo" propriamente dito. Há uma série de vetores previamente classificados (instâncias) e um novo dado é classificado a partir dos mais parecidos com ele.

Os modelos mais utilizados que se baseiam em instâncias são o *nearest-neighbor* e o K *nearest-neighbor* (KNN), o qual é mais indicado para quando temos uma base de treino muito grande (Witten *et al*. 2011). A diferença entre os dois é que, enquanto o classificador *nearest-neighbor* utiliza o vizinho mais próximo para classificação, o k *nearest-neighbor* (KNN) utiliza os k vizinhos mais próximos. Como mostram Groth e Munterman (2011) e já apresentado anteriormente, no KNN não há uma etapa de criação de um modelo. Nesse tipo de classificador, a classe do documento é definida a partir dos documentos mais similares já classificados. São selecionadas as k instâncias mais próximas para realizar a classificação. A definição do número k a ser utilizado é crítica, pois caso se utilize um número muito grande, os vizinhos podem incluir pontos de outras classes. E caso seja muito pequeno, pode ser muito sensível. Pelo fato desse modelo não necessitar de ter uma separação linear entre as classes, Feldman e Sanger (2007) atribuem a esse classificador uma grande robustez.

Existem diversas maneiras de definir os critérios para se calcular o k. A abordagem mais simples utiliza o número de termos comuns entre os dois documentos e faz uma

normalização entre os dois documentos com base na distância euclidiana. O coeficiente de Jaccard leva em consideração a relação entre a interseção dos termos e a união dos termos dos dois documentos. Portanto, um algoritmo de KNN, primeiro calcula a similaridade da notícia que se deseja classificar com as demais notícias da base do treino, seleciona as k instâncias mais próximas e escolhe a classe mais freqüente dentre os k vizinhos. A distância euclidiana pode ser calculada com base na Equação 9:

$$D\ (x,y) = \sqrt{(x_1 - y_1)^2 + \ (x_2 - y_2)^2 + \ \ldots + \ (x_n - y_n)^2} \tag{9}$$

A distância utilizando o coeficiente de Jaccard é calculada de acordo com a Equação 10:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{10}$$

Uma variação do KNN, porém menos utilizada, é o *distance-weighted nearest neighbors*, onde os vizinhos mais próximos recebem maior influência na classificação. Nesse tipo de classificador podem ser consideradas todas as instâncias ou somente as k mais próximas. O cálculo da distância utilizando os pesos é dado por:

$$D\ (x,y) = \sqrt{w_1^2(x_1 - y_1)^2 + \ w_2^2(x_2 - y_2)^2 + \ \ldots + \ w_n^2(x_n - y_n)^2} \tag{11}$$

Onde w_1, w_2,..., n são os pesos de cada atributo, os quais são atualizados a cada nova instância treinada. Witten *et al*. (2011) explicam que a atualização dos pesos é feita com base nos erros e acertos do classificador. A correta classificação aumenta o peso do atributo, e a classificação incorreta reduz o peso do atributo.

3 REVISÃO DA LITERATURA

A revisão de literatura tem por objetivo identificar estudos semelhantes a este realizado, seja por trabalhar com variação de parâmetros de pré-processamento, por trabalhar com mineração de textos utilizando notícias ou por criar classificadores textuais para depois relacionar com a bolsa de valores, enfim, levantar o estado da arte.

O experimento de Schumaker *et al*. (2009) combinou *bag-of-words, noun phases e named entities* para pré-processar as notícias. O método de seleção baseou-se na ocorrência mínima de palavras por documento e com isso foi classificada a notícia em positiva, negativa ou neutra. Foram coletados também dados do valor das ações da S&P 500, minuto a minuto, a partir de 60 minutos antes da divulgação da notícia e com isso feita uma regressão linear prevendo o valor para os próximos 20 minutos. As informações resultantes da regressão e de *text mining* de Schumaker *et al*. (2009) são jogadas para uma base de dados e depois realizada uma regressão utilizando *support vector machine* que fornece o valor esperado para o valor da ação nos próximos 20 minutos baseado na notícia divulgada. Para o experimento, foram coletadas notícias e o valor das ações durante 5 semanas. Feito isso, foi feita uma simulação com investimento de mil dólares e gerou retorno médio de 1,59% utilizando *bag-of-words*, 2,57% com *noun phrases* e 2,84% com *named entities*.

Mittermayer (2004) desenvolveu um programa chamado NewsCATS que durante a extração trabalhou com N-Gram e para classificar as notícias em três grupos (boas, ruins ou sem alteração), utilizou SVM. A base de dados foi restrita para empresas com volume de negócios superior a 5 milhões de dólares buscando obter ações com grande liquidez. As notícias foram coletadas somente durante o período de funcionamento das NYSE e NASDAQ. Nesse estudo de Mittermayer, foram utilizadas mais de seis mil notícias.

Diferente dos autores anteriores que utilizaram o SVM como classificador, Gidófalvi (2001) classificou as notícias relacionadas a doze ações listadas na NASDAQ utilizando o classificador Naïve Bayes para agrupá-las em três grandes grupos (alta, baixa e indiferente) e analisar o comportamento das ações 20 minutos antes e 20 minutos depois da publicação das notícias. Como o estudo baseou-se somente em analisar a variação de curto prazo, foram excluídas notícias publicadas fora do horário de encerramento da bolsa, assim como o estudo realizado por Mittermayer (2004).

Como o SVM é um classificador binário, Fung *et al*. (2002) o utilizou para classificar as notícias com tendência de aumento ou queda de preço de 614 ações do mercado de Hong Kong. Em casos onde a classificação era conflitante (quando os dois classificadores eram

classificados como positivo ou negativo), o sistema retornava a informação de que não havia recomendação. Foram coletadas aproximadamente 350 mil notícias e aproximadamente 2000 valores de preço para cada ação durante seis meses. Os dados dos cinco primeiros meses foram utilizados para a montagem do modelo de previsão e o ultimo mês para teste.

Das e Chen (2007), diferentemente de Fung *et al.* (2002), trabalharam com a coleta de mensagens de investidores em fóruns virtuais para criação dos classificadores. As mensagens foram classificadas em otimistas, pessimistas ou neutras, utilizando para isso, 5 classificadores diferentes, que trabalhavam com *bag-of-words, noun phrasing* e *Naive Bayes*. Além da base das mensagens coletadas por um programa específico para esse fim e do valor das ações, foi utilizado um dicionário da língua inglesa, um léxico construído com palavras do jargão financeiro e uma gramática para definir o conjunto de regras. Foram estudadas 35 ações do Morgan Stanley High-Tech Index (MHS35) e também diferentemente dos demais estudos apresentados, o sentimento era definido diariamente, ou seja, cada comentário classificado como positivo acumulava um ponto e cada comentário negativo retirava um ponto. Mensagens nulas não pontuavam e ao final do dia chegava-se ao valor do dia com a soma da pontuação de cada mensagem. O estudo foi feito com dados de 88 dias úteis e mais de 397 mil mensagens, com média diária de 4500 comentários dos investidores. Uma conclusão relevante do estudo foi que em média, as mensagens refletem a variação de preços com um atraso médio de 50 minutos. Também ao final do dia, era feito uma análise do sentimento agregado do índice estudado e conclui que o sentimento agregado gerava melhores retornos do que a análise individual de cada ação.

Li (2009) categorizou manualmente 30.000 comentários referentes ao mercado financeiro com relação ao sentimento em positivo, negativo, neutro ou incerto e ao conteúdo, referiam-se a liquidez, rentabilidade, operações, etc. Essas sentenças foram utilizadas como dados de treinamento classificador *Naive Bayes* e, em seguida, esse classificador foi utilizado para categorizar o sentimento e conteúdo de mais de 13 milhões de declarações e mostrou a relação do sentimento dos comentários com o retorno positivo do investimento.

Wuthrich *et al*. (1998) foram um dos primeiros a utilizarem ferramentas de *machine learning* para prever o preço das ações. Foram utilizadas notícias disponibilizadas na internet para prever o movimento das ações dos mercados americano, europeu e asiático. Foi utilizado um dicionário de termos da área e *bag-of-words* durante o pré-processamento. O classificador que obteve melhores resultados foi o KNN, com n igual a 9. Foi feita uma simulação para três meses e na média de todos os mercados, obteve-se ganho de 5,2% enquanto a média de crescimento do mercado foi de 1,48%.

Butler e Keselj (2009) fizeram seu estudo baseado em relatório anuais das empresas para prever quais iriam ter performance melhor ou pior que o índice S&P 500 no ano seguinte. Ou seja, ao contrario de alguns estudos já mostrados que analisam o impacto das notícias no curto prazo, Butler e Keselj trabalham com previsões de longo prazo. Eles utilizaram *n-gram* durante o pré-processamento. O *n-gram* foi aplicado tanto para grupo de palavras quanto de caracteres e em seguida utilizado SVM para classificar a performance do ano seguinte.

Em um estudo mais recente, Kim e Kim (2014) utilizaram mensagens de investidores publicadas no *Yahoo! Finance* para identificar o sentimento e prever o valor das ações. Foram coletados mais de 32 milhões de mensagens referentes a 91 empresas. As mensagens foram classificadas em "*strong buy", "buy", "strong sell", "sell" e "hold*" e depois aplicada em um algoritmo de *Naive Bayes*. Não foi possível concluir que é possível prever a variação das ações com base nos sentimentos dos investidores. Foi concluído que em geral, o sentimento do investidor é moldado pelo comportamento do mercado.

Smailovic *et al*. (2014) estudaram a relação entre os sentimentos expressos em publicações do *Twitter* e a variação dos preços das ações das empresas mencionadas. As notícias foram pré-processadas variando os parâmetros buscando identificar a melhor configuração de pré-processamento para treinar um classificador SVM. Foram alterados os parâmetros de *tokenization, stemming, stopwords*, frequência de termos e TF e IDF. O melhor resultado foi atingido utilizando TF, *n-gram* máximo igual a 2 e frequência mínima de termos igual a 2. Foram testados três classificadores: *Naive Bayes*, SVM e KNN. *Support Vector Machine* foi o que obteve melhor performance.

Dumais *et al*. (1998) comparou cinco tipos de classificadores em função do tempo de aprendizado, tempo para classificação e acurácia dos classificadores criados a partir de uma base de notícias extraídas da Reuters. Os cinco classificadores utilizados foram: *Find Similar, Naive Bayes*, Redes Bayesianas, Arvores de decisões, e SVM. Durante o pré-processamento, foi utilizado TF e IDF, frequência mínima de termos, *bag-of-words, noun phrases* e *n-gram*. Dumais *et al*. (1998) concluíram que a utilização de *bag-of-words* gerou melhores resultados que *n-gram*. E o classificador que obteve melhores resultados foi o SVM, pois é rápido para treinar, rápido para classificar novas notícias e com alta acurácia.

O estudo de Joachims (1998) consistiu em mostrar tanto teoricamente quanto empiricamente que SVM gera melhores resultados que os demais classificadores para cetegorização textual. Para isso, comparou com Naive Bayes, C4.5, *K-nearest neighbors e Rochio Algorithm* utilizando notícias da Reuters como base, assim como Dumais *et al*. (1998).

A seleção de atributos foi feita utilizando *information gain*. O classificador que obteve piores resultados foi o C4.5, enquanto que conseguiu melhores rendimentos foi o SVM, seguido do KNN.

Siolas e d'Alche-Buc (2000) utilizaram vinte bases de notícias, cada uma contendo mil notícias de temas distintos para comparar SVM e KNN. A seleção de atributos foi realizada de acordo com a proximidade semântica dos termos. Foi concluído que o aumento do número de atributos não gerava melhores resultados, além de que o SVM resultou em melhor acurácia para todas as métricas medidas.

Além dos modelos SVM e KNN utilizados por Siolas e d'Alche-Buc (2000), Yang e Liu (1999) também compararam algoritmos de redes neurais, Naive Bayes e Linear Least Square Fit (LLSF), utilizando a mesma base de notícias da Reuters utilizada por Joachims (1998) e Siolas e d'Alche-Buc (2000). As conclusões foram que SVM, LLSF e KNN apresentaram melhores resultados que Naive Bayes e Redes neurais.

A Tabela 1 resume as ferramentas utilizadas para pré-processamento e as ferramentas utilizadas para criação dos classificadores (*machine learning*). A tabela foi elaborada a partir do estudo de Hagenau *et al*. (2013). Os campos que estão em branco ocorrem pelo fato do artigo não abordar a respectiva etapa do processo.

Tabela 1 - Resumo de ferramentas utilizadas em estudos relacionados.

Autor	Pré-processamento e seleção de atributos	Classificador de notícias
Mittermayer (2004)	N-Grama, TF, IDF	SVM
Schumaker (2009)	bag-of-words, noun phrases, named entities, frequência minima	SVM com regressão
Fung *et al*.. (2002)		SVM
Gidófalvi (2001)		Naive Bayes
Das e Chen (2007)	bag-of-words e noun phrases	Naive Bayes
Li (2009)		Naive Bayes
Groth *et al*.. (2009)	bag-of-words, stopwords	SVM
Wuthrich *et al*.. (1998)	bag-of-words	Naive Bayes e KNN
Antweiler *et al*. (2004)	bag-of-words	Naive Bayes e SVM
Tetlock *et al*.. (2008)	bag-of-words	Rácio de palavras negativas
Butler e Keselj (2009)	N-grama	SVM
Kim e Kim (2014)		Naïve Bayes

Smailovic *et al*.. (2014)	Stopwords, stemming, n-gram, frequencia dos termos, TF e IDF	KNN, Naive Bayes e SVM
Dumais *et al*.. (1998)	TF e IDF, frequência mínima de termos, bag-of-words, noun phrases e n-gram	Find Similar, Naive Bayes, Redes Bayesianas, Árvores de decisões, e SVM
Joachims (1998)	Information gain, IDF	SVM, Naive Bayes, KNN, Rochio Alogorithim e C4.5
Siolas e d'Alche-Buc (2000)	Proximidade semântica	SVM, KNN
Yang e Liu (1999)		SVM, KNN, Rede neural, LLSF, Naïve Bayes

Fonte: Adaptado de Hagenau et al. (2013)

Por conta da dificuldade de comparação da eficiência dos classificadores dos diferentes estudos, esse aspecto não foi abordada nessa revisão. É possível perceber que os classificadores mais utilizados para classificar notícias ou outros conteúdos textuais relacionados à ações são SVM, *K-nearest neighbors* e *Naive Bayes*. A provável justificativa para utilização do SVM é seu grande poder de aprendizado mesmo quando utilizado com base de dado grande. Feldman e Sanger (2007) reforçam essa teoria quando afirmam que o SVM é um algoritmo muito rápido efetivo para classificação de texto. Já o Naive Bayes pode ser explicado pelo seu módulo específico para processamento de bases textuais. Quando analisado o tipo de pré-processamento, apesar da maior variedade, o mais utilizado é o *bag-of-words*. É possível perceber também, que dentre os três classificadores mais utilizados, Naive Bayes é aquele que apresenta os piores resultados nos testes empíricos, enquanto *Support Vector Machine* obtém melhor performance que os demais. Outro ponto importante de ser destacado é que todos os estudos semelhantes identificados foram realizados para língua inglesa. Para esse estudo, porém, foi adotada a língua portuguesa como padrão para a coleta das notícias. Dessa forma, o trabalho também será útil para ser confrontado com os resultados dos demais estudos a fim de verificar se as melhores técnicas identificadas para a língua inglesa também se aplicam ao português.

4 METODOLOGIA

A metodologia para desenvolvimento deste trabalho foi baseada em uma prévia revisão bibliográfica para fornecer embasamento teórico e um estudo de caso para validar a teoria. De acordo com Chizzotti (2006), o estudo de caso é uma caracterização abrangente para designar uma diversidade de pesquisas que coletam e registram dados de um caso particular, ou de vários casos, a fim de organizar um relatório ordenado e crítico de uma experiência, ou então para avaliá-lo analiticamente. Yin (2001) também concorda que o estudo de caso é uma técnica de pesquisa abrangente, mas para ele, assim como outras formas de pesquisa, o estudo de caso representa uma maneira de se investigar um tópico empírico seguindo-se um conjunto de procedimentos pré-especificados. Assim como Chizzoti (2006), Ventura (2007) entende que ao realizar um estudo de caso, o autor visa à investigação de um caso específico, mas completa a definição dizendo que para se caracterizar como estudo de caso, o conteúdo da investigação deve ser bem delimitado, contextualizado em tempo e lugar para que se possa realizar uma busca detalhada de informações.

O estudo do projeto foi baseado na empresa Vale SA, segunda maior mineradora do mundo, segundo dados da própria empresa, fundada em 1942, hoje está presente no Brasil e em mais trinta países. Trata-se de um estudo de caso observativo e não participativo, ou seja, não houve contato direto com a empresa, com o intuito de criar classificadores automáticos de notícias construídos a partir das notícias divulgadas em sites de notícias. O estudo foi dividido em cinco etapas conforme mostra Figura 6. A primeira compreendeu uma ampla revisão bibliográfica e de literatura para embasamento teórico e foi realizada durante todo o projeto e de maneira paralela às demais etapas. A revisão contemplou o estudo de artigos científicos, livros relacionados e publicações acadêmicas relacionadas à *machine learning, text mining, natural language processing, big data*, mercado acionário, fatores que impactam o valor das ações e estudos relacionados.

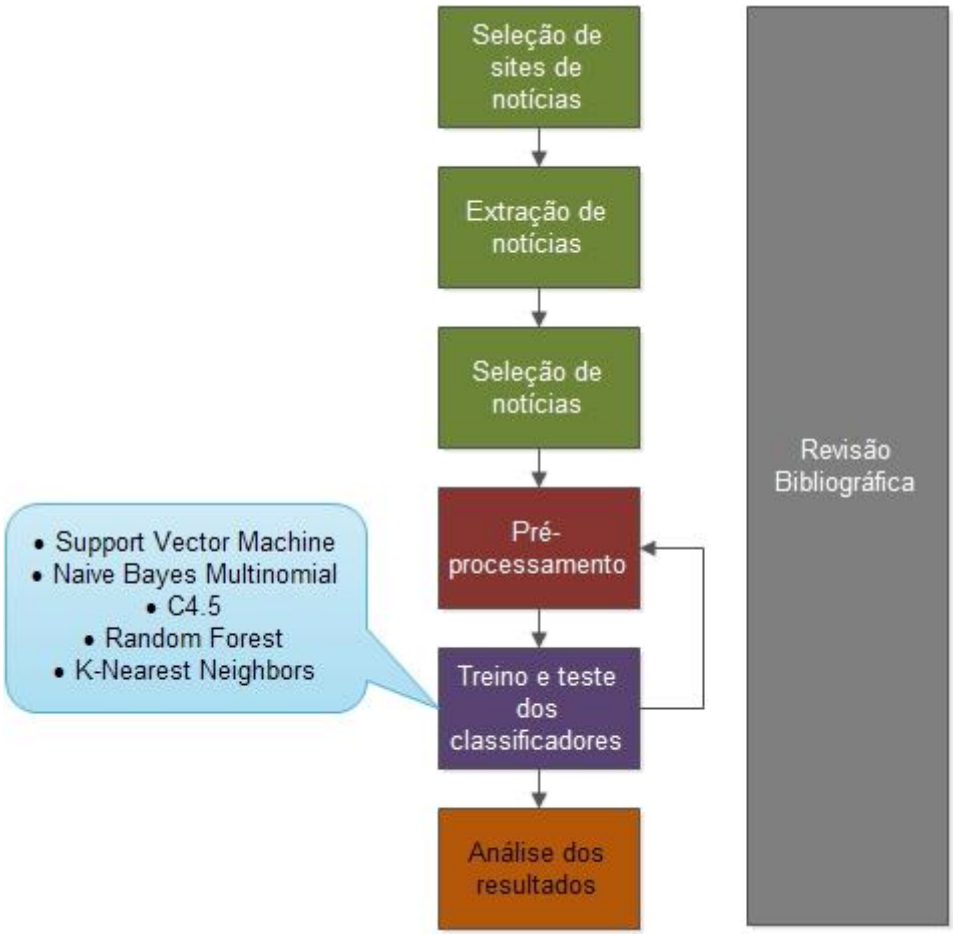

Figura 6 - Fluxograma metodologia
Fonte: Elaborado pelo autor

A segunda etapa correspondeu à coleta e seleção das notícias. Foram selecionados os sites de notícias para posterior seleção das notícias relacionadas à Vale S.A. e ao mercado de mineração. Após a seleção dos sites para coleta de notícias, foi realizada a extração das notícias utilizando o *software Web Content Extractor*, que rastreia e extrai as informações da internet. Foi feita uma análise preliminar de forma manual com todas as notícias de maneira a manter somente aquelas que possuíam relação direta com o tema a ser analisado.

A terceira etapa é a de pré-processamento da notícia utilizando ferramentas e técnicas de processamento de linguagem natural (PLN). O pré-processamento permite uma primeira estruturação dos dados (Gomes, 2012). A primeira atividade do pré-processamento foi a classificação manual de todas as notícias para servir de subsídio para o treino do classificador. Para o estudo, foi utilizada uma classificação binária para as notícias (positivo ou negativo). Em seguida foi feita a segmentação do texto em palavras (*tokenization*). Foram alterados os parâmetros da tokenização para verificar o impacto dessas mudanças no resultado dos classificadores. Buscando reduzir a quantidade de palavras contidas no texto, foram

removidas as palavras chamadas de *stop words*, que são normalmente, palavras que não adicionam valor ao texto. Após a segmentação de palavras e remoção de *stop words*, cada documento pré-processado ficou representado em formato de uma matriz tabela que representa se um conjunto de palavras está contido ou não no documento. Durante a tokenização, os atributos inicialmente nominais (palavras) são transformados em variáveis binárias (0,1) e entendidos como números pelos classificadores. Com essa transformação, a base que inicialmente era desestruturada, se torna estruturada e possível de ser analisada. A base então é dividida em base de treino e de teste. A base de treino é utilizada para a criação do classificador que posteriormente é aplicado na base de teste para validação da sua performance.

Durante a fase de treino, os classificadores criam padrões baseados na presença de palavras e sua classificação preliminar. Com o classificador treinado, ele é testado utilizando a base de treino para verificar seu desempenho. Ele recebe, então, um *input* (notícia pré-processada) e define em qual das n classes ele pertencerá e em seguida é confrontado com a classificação manual realizada para verificar o seu resultado. Essa etapa de processamento das notícias com a utilização de algoritmos de *Machine Learning* para criação de classificadores é resumida na Figura 7.

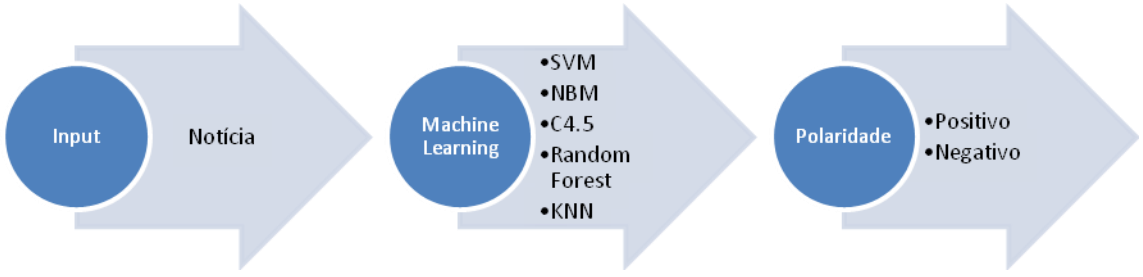

Figura 7 - Processamento das notícias utilizando *Machine Learning*
Fonte: Elaborado pelo autor

Buscando identificar o classificador que gera melhores resultados para o tema e os melhores parâmetros de pré-processamento, foram utilizados cinco modelos: *Support Vector Machine*, *Naive Bayes Multinomial*, C4.5, *Random Forest e K Nearest Neighbor* e alterados diversos parâmetros de tokenização e seleção de atributos, que serão explicados em detalhes no capítulo seguinte. Para cada variação dos parâmetros do pré-processamento, foram registrados os resultados do classificador para posterior análise e comparação. Além da variação dos parâmetros de tokenização, foram alterados a frequência mínima dos termos e

testados alguns métodos de seleção de atributos para verificar qual melhor se adequava e gerava ganho de performance dos classificadores.

A última etapa do trabalho abarcou a comparação entre os classificadores treinados a fim de verificar qual melhor se aplica para o contexto analisado e identificar as palavras que melhor auxiliam na classificação da polaridade da notícia. Como alertaram Feldman e Sanger (2007), para os resultados poderem ser comparados eles devem seguir algumas condições: serem trabalhados sob a mesma base de dados e utilizarem os mesmos indicadores de performance. Todos os testes foram realizados utilizando a mesma base de notícias. A comparação foi realizada utilizando seis indicadores: acurácia, *precision*, *recall*, *F-measure*, Área ROC e estatística Kappa. A acurácia é a métrica mais simples para medir o desempenho do classificador e representa a relação entre o número de classificações corretas e o número total de classificações e é definida pela Equação 12:

$$Acurácia = \frac{Número\ de\ classificações\ corretas}{Numero\ de\ classificações} \quad (12)$$

A curva ROC (*receiver operating characteristic*) é um gráfico que representa a sensibilidade do classificador calculada pela taxa de verdadeiros positivos, ou seja, notícias que previamente foram classificadas como positivas e que o classificador classificou como positivas e a taxa de falsos positivos, que representam as notícias que foram classificadas previamente como positivas e após a construção do classificador, foram definidas como negativas. A área abaixo da curva ROC representa a área ROC e terá seu valor entre 0 e 1.

Comparando e observando as duas curvas ROC na Figura 8, é possível perceber que o classificador perfeito seria aquele onde a linha da curva ROC seria horizontal com y igual a 1. Analogamente, quanto mais próximo dessa linha perfeita, melhor é o modelo e consequentemente maior a área ROC. É possível concluir então, que quando comparados dois modelos, aquele que possui maior área ROC é portanto, o que apresenta menor índice de falsos positivos. Logo, tanto quanto maior a área ROC, melhor representado está o nosso classificador.

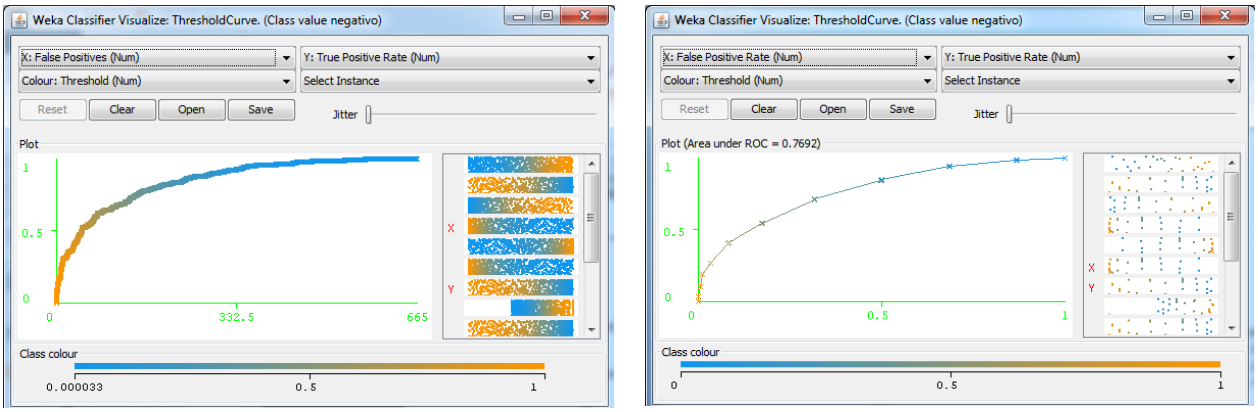

Figura 8 Comparativo curva ROC
Fonte: Elaborado pelo autor.

Já os indicadores de *precision e recall*, geralmente são utilizados de maneira conjunta. *Precision* para as notícias positivas mede o número de notícias classificadas corretamente como positivas dividido pelo total de notícias classificadas como negativas. Já a medida de *recall* para as notícias positivas mensura número de notícias classificadas corretamente como positivas dividido pelo total de notícias inicialmente classificadas como positivas. Assim como calculados para as notícias positivas, *precision* e *recall* também podem ser mensurados para as notícias negativas.

$$\text{Precision} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos positivos}} \tag{13}$$

$$\text{Recall} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}} \tag{14}$$

A partir desses dois indicadores, é possível criar outra métrica a partir da média harmonica entre os dois, chamada de *F-measure* ou medida F.

$$F - measure = \frac{2 * recall * precision}{recall + precision}$$

$$= \frac{2 * Verdadeiros\ positivos}{2 * Verdadeiros\ positivos + Falsos\ positivos + Falsos\ negativos} \tag{15}$$

Como a medida F é baseada na *recall e precision*, para a comparação e análise dos resultados, essas duas serão omitidas.

A estatística Kappa mede o grau de concordância entre os valores previstos e os observados e a probabilidade aleatória de acerto. Ou seja, mede ao mesmo tempo, confiabilidade e precisão do classificador.

$$Kappa = \frac{\mathrm{P(A)} - \mathrm{P(E)}}{1 - P(E)} \tag{16}$$

Onde P(A) representa a probabilidade de acerto do classificador (acurácia) e P(E) mede a probabilidade de acerto do classificador atribuída ao acaso.

O cálculo de P(E) pode ser feito a partir da matriz de confusão que está representada na Tabela 2

Tabela 2 - Matriz de confusão

		Valor Verdadeiro		
		Positivo	Negativo	Soma
Valor previsto	Positivo	Verdadeiro positivo (VP)	Falso Positivo (FP)	**VP + FP**
	Negativo	Falso negativo (FN)	Verdadeiro Negativo (VN)	**FN + VN**
	Soma	**VP + FN**	**FP + VN**	

Fonte: Elaborado pelo autor

$$P(E) = \frac{VP + FP}{N° \, de \, amostras} * \frac{VP + FN}{N° \, de \, amostras} + \frac{FN + VN}{N° \, de \, amostras} * \frac{FP + VN}{N° \, de \, amostras} \tag{17}$$

O Kappa varia entre 0 e 1, e quanto mais próximo de 1, mais confiável e preciso é o modelo e quanto mais próximo de 0, mais aleatório está o classificador.

As principais etapas do trabalho podem ser resumidas na Figura 9.

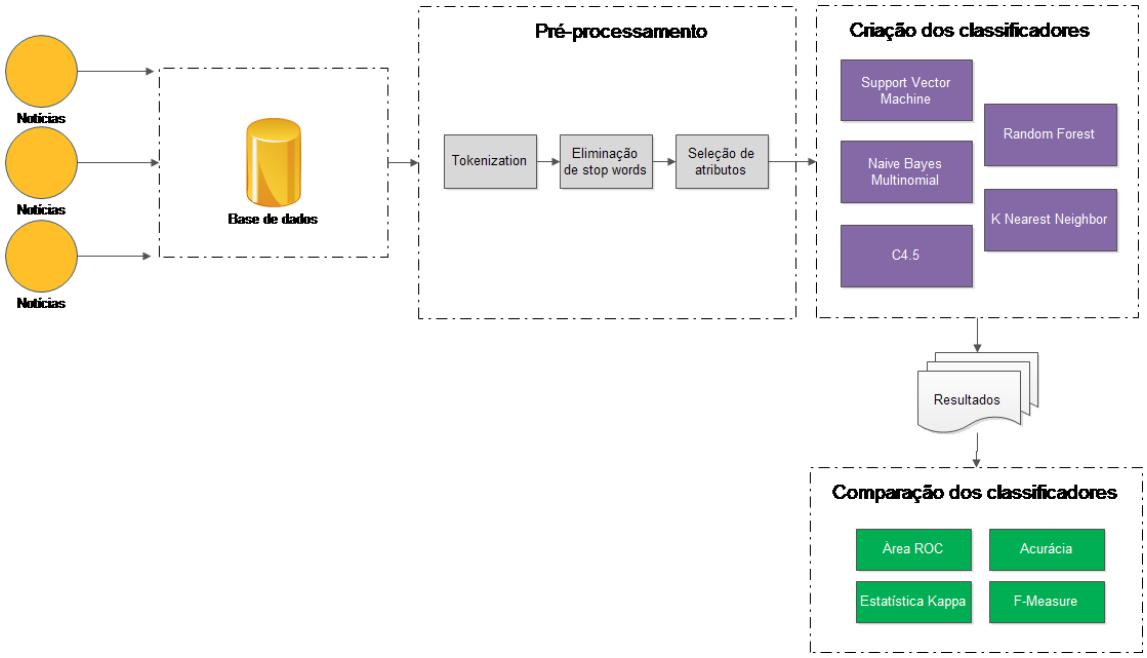

Figura 9 - Etapas da metodologia para desenvolvimento do trabalho
Fonte: Elaborado pelo autor

Conforme mostra a Figura 9, a primeira etapa do projeto, após a revisão bibliográfica, foi a criação do banco de dados com a seleção e extração das notícias. A segunda etapa consistiu em variar os parâmetros de pré-processamento, para na etapa seguinte, serem criados os cinco classificadores para cada mudança de pré-processamento. A última etapa foi a comparação dos diferentes classificadores e parâmetros de pré-processamento a fim de identificar aqueles que obtinham melhores resultados para a base em estudo.

5 ESTUDO DE CASO

O estudo de caso foi realizado utilizando informações da Vale SA, fundada em 1.942 e segundo dados empresa e do mercado, segunda maior mineradora do mundo, segundo dados da própria empresa. Hoje está presente no Brasil e em mais trinta países. Trata-se de um estudo de caso observativo e não participativo, onde todas as informações da empresa necessárias para execução do processo foram extraídas da internet, conforme explica seção 5.1. Foram coletadas 1.657 notícias relacionadas à empresa para então serem pré-processadas de diferentes maneiras e inseridas em algoritmos de *machine learning* que treinaram e testaram um modelo de classificação de notícias. Foi utilizado cinco modelos de classificadores e depois comparados seus resultados a fim de identificar àquele que gera melhores resultados para a base em estudo. A comparação foi feita utilizando indicadores de resultado dos classificadores que avaliaram sua acurácia, precisão, aleatoriedade, etc.

5.1 COLETA DE NOTÍCIAS

Para a primeira etapa da execução do trabalho, a coleta das notícias, foi escolhido o *software web content extractor* utilizado em sua versão *trial*. Esse tipo de *software* é conhecimento como *webcrawler*, ou rastreador web, que consegue navegar na web de maneira automatizada. Um *crawler* consegue visitar várias paginas para coletar informações, seguir *hiperlinks* e realizar download do conteúdo desejado (Liu, 2009). O primeiro passo para a coleta das notícias pelo *webcrawler* foi a seleção dos sites de notícias. Como a busca consistiu inicialmente identificar os campos padrões de notícias, para a extração do conteúdo, foram restritos somente a sites com campos de título, data e hora padronizados para todas as notícias. Outro critério utilizado para a seleção das notícias foi estar no idioma português. Diante dessas restrições, foram selecionados os sites:

- Agência Brasil - http://agenciabrasil.ebc.com.br/
- Bol - http://www.bol.uol.com.br/
- Epoca Negócios - http://epocanegocios.globo.com/
- Estadão - http://www.estadao.com.br/
- Folha de SP - http://www.folha.uol.com.br/
- IG - http://www.ig.com.br/

- R7 - http://www.r7.com/
- Terra - http://www.terra.com.br/
- Último Segundo - http://ultimosegundo.ig.com.br/
- Vale - http://saladeimprensa.vale.com/pt/noticias/index.asp

Para todos os sites, foi utilizado como argumento de busca “Vale mineração”, pois dessa forma foram retornados tanto notícias relacionadas à Vale S.A quanto notícias do setor. Selecionados os sites de notícias e o argumento de busca, o link com as notícias é inserido no *Web Contect Extractor* que reconhece todas os elementos similares, que no caso são as notícias.

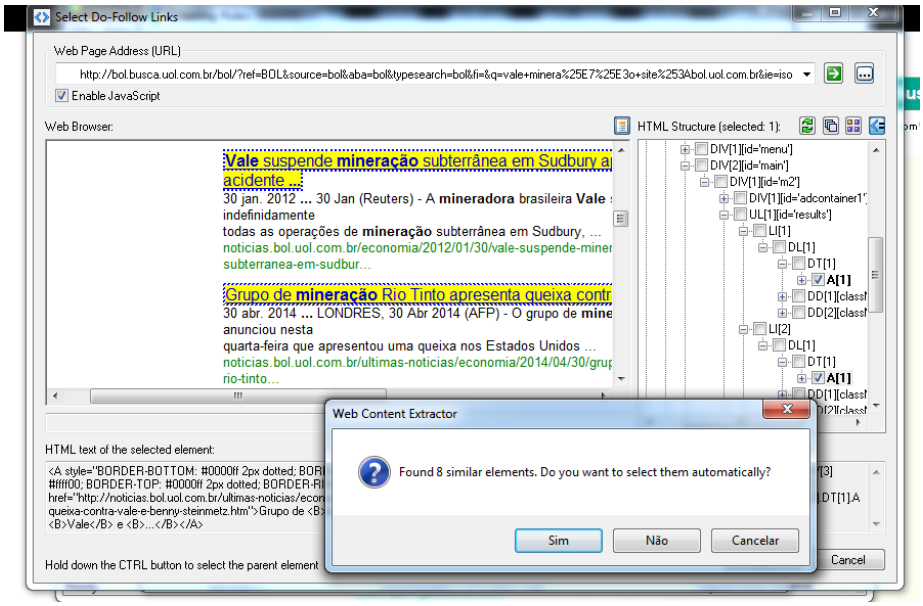

Figura 10 - *Webcrawler* identificando campos em comum
Fonte: Elaborado pelo autor

Reconhecidas as notícias, selecionaram-se os campos que desejava extrair as informações. Foi selecionado o título da notícia (o qual será feita a análise da polaridade), a data a qual a notícia foi publicada, hora de publicação, e o primeiro parágrafo. O primeiro parágrafo foi coletado para que chegasse a uma conclusão em relação à polaridade da notícia nos casos onde não possível identificar se a notícia era positiva ou negativa somente com seu título.

As Figuras 11, 12 e 13 mostram o reconhecimento dos campos pelo *crawler*.

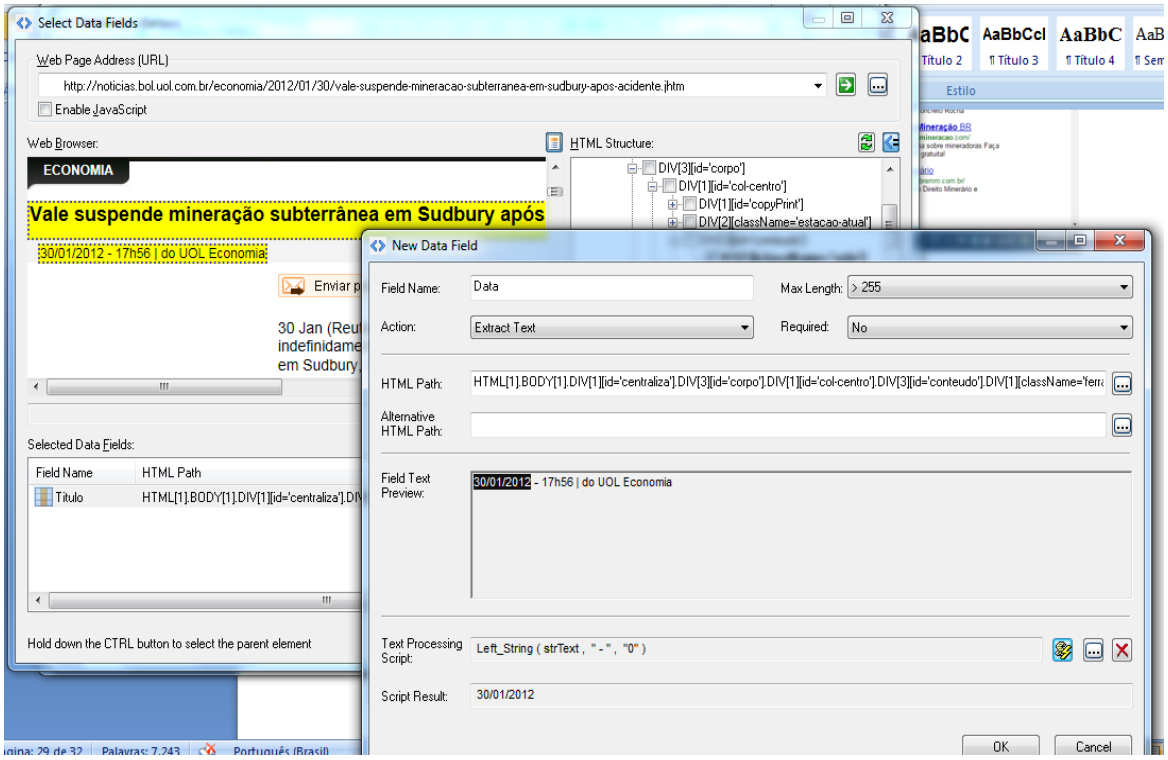

Figura 11 - Definição das informações a serem extraídas (data)
Fonte: Elaborado pelo autor

A definição dos campos a serem extraídos é importante para criação de uma tabela estruturada que possa ser processada posteriormente. Para o processamento do texto, foi identificado que seriam necessários a extração das informações do título, a data de publicação da notícia, a hora e o primeiro parágrafo. A Figura 11 representa o reconhecimento do campo data. Na imagem é possível perceber que a data, a hora e a origem da notícia estão presentes no mesmo campo. O software consegue fazer a extração dessas três informações de maneira separada utilizando um *script*. A Figura 12 mostra o reconhecimento do campo hora.

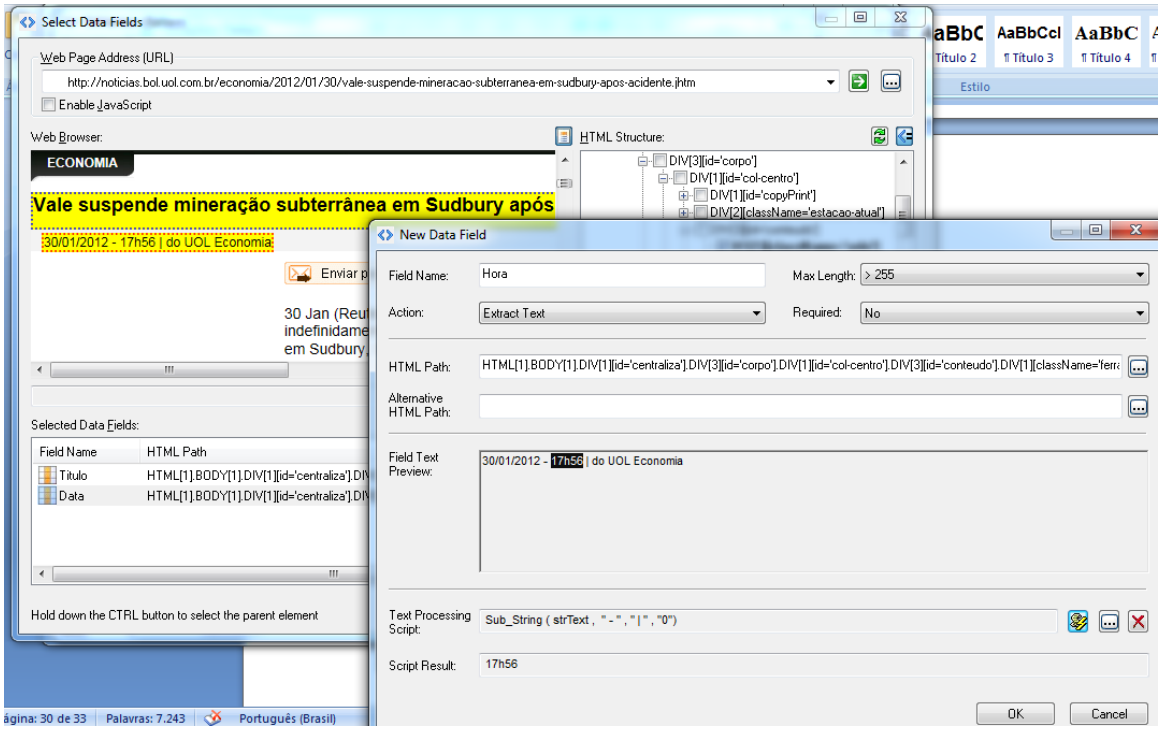

Figura 12 - Definição das informações a serem extraídas (hora)
Fonte: Elaborado pelo autor

Como mencionado anteriormente, a informação da data estava junta da hora para o exemplo mostrado. Mais uma vez foi utilizado o *script* do *software* para extrair somente a informação da hora. A separação dos dois campos é importante, pois em outros sites esses dois campos geralmente aparecem separados. Como posteriormente as bases de todos os sites foram unificadas, os campos deveriam ser semelhantes de modo que não houvesse conflito ou perda de informação. A Figura 13 mostra a extração do último atributo da base de notícias, que é o primeiro parágrafo.

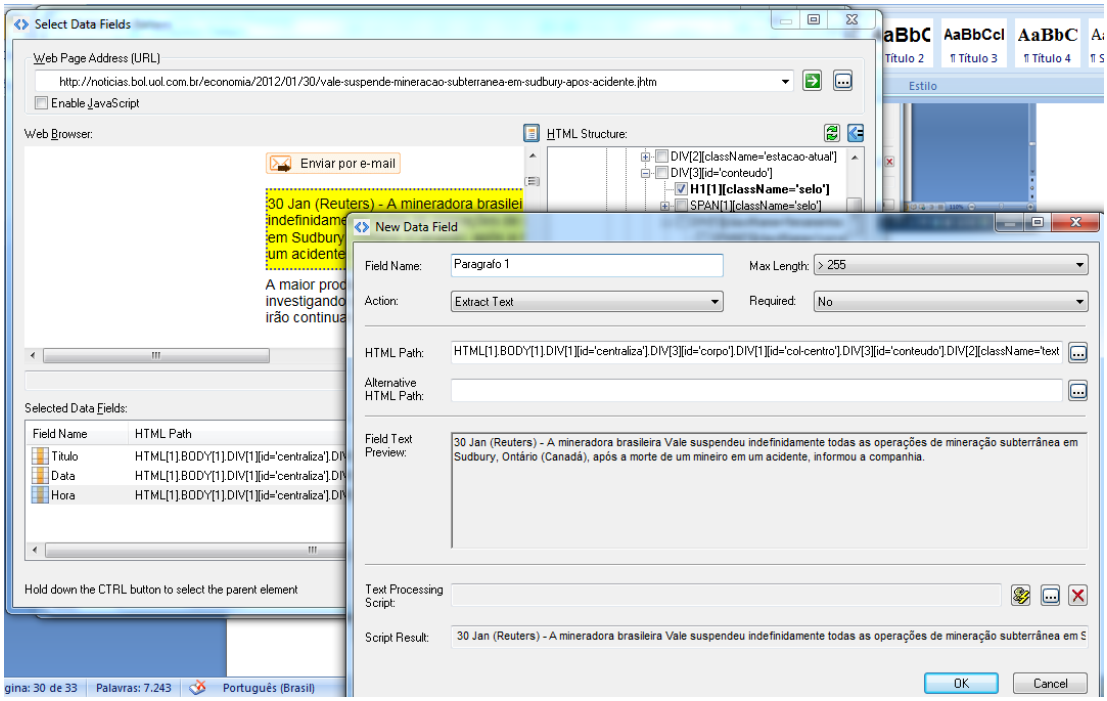

Figura 13 - Definição das informações a serem extraídas (parágrafo 1)
Fonte: Elaborado pelo autor

Cabe ressaltar que as informações do primeiro parágrafo foram extraídas somente por questões de segurança. Como em alguns casos não é possível tirar conclusões quanto à polaridade da notícia somente com as informações utilizadas no título, tinha-se uma base de informações extras para o auxílio na classificação da polaridade. Após a seleção do site e dos parâmetros a serem extraídos, tem-se o inicio da extração das informações desejadas de maneira automática pelo sistema.

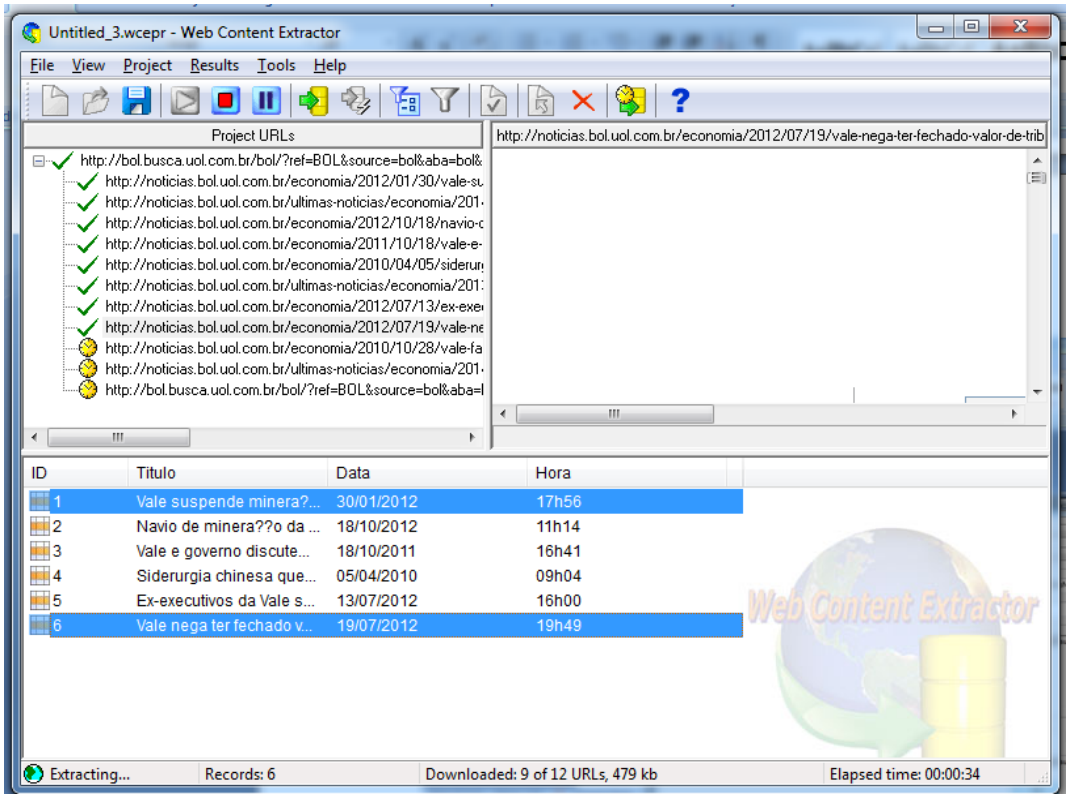

Figura 14 - *Crawler* realizando a extração das notícias
Fonte: Elaborado pelo autor

Ao término do rastreamento e extração das informações das notícias, é possível exportar os dados coletados para uma planilha em Excel ou banco de dados do Access. Esse procedimento de coleta de notícias foi repetido para todos demais sites selecionados anteriormente. Além da possibilidade de exportar para Excel ou Access, é possível converter as informações extraídas para outros bancos de dados, como SQL, MySQL ou ODBC, como mostra a Figura 15. Para o trabalho, todas as notícias foram exportadas para o Excel.

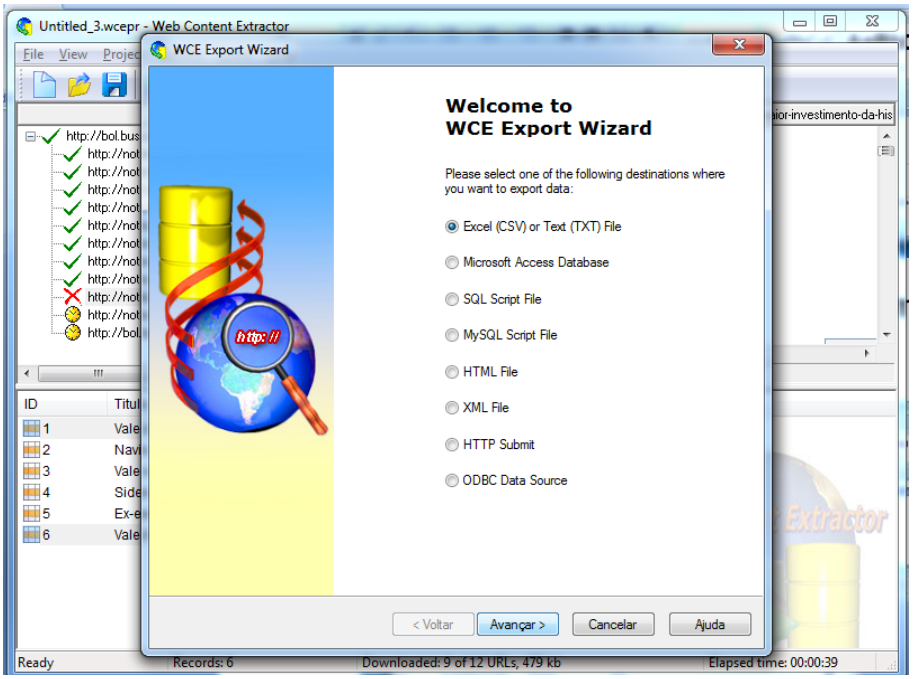

Figura 15 - Exportação das notícias para banco de dados
Fonte: Elaborado pelo autor

Foram coletadas ao todo, 1.657 notícias divulgadas desde 21 de fevereiro de 2001 até 04 de setembro de 2014. Porém, foi percebido que apesar da definição do argumento de busca antes da coleta das notícias, muitas delas não possuíam relação com o tema a ser analisado. Portanto, foram analisadas todas as notícias e as que não interessavam para análise e poderiam atrapalhar o modelo, foram excluídas. Segue abaixo um exemplo de notícia extraída da Folha de SP publicada no dia 25 de agosto de 2014 às 8h14.

"Defensor do empresariado nacional, Ermírio criticou a ditadura e tentou carreira política."

Analisando o título, é possível concluir que não há nenhuma relação com o tema do estudo e argumento de busca. Existem duas hipóteses para esse tipo de notícia estar contido no resultado da extração. A notícia estar relacionada com a Vale SA, mas não é possível ver essa relação no título da notícia ou falha na busca que retornou uma notícia sem relação com o argumento. Após a remoção das notícias que não interessavam, sobraram à base, 1.138 notícias.

5.2 CRIAÇÃO DE CLASSIFICADORES

Finalizada a extração das notícias e remoção das que não continham relação com o conteúdo desejado, foi feita a classificação de maneira manual em positivo ou negativo, a

partir do título e do primeiro parágrafo, quando não era possível obter conclusões a respeito da polaridade da notícia somente a partir do título.

Como regra geral, foi definido que quando a notícia referenciava um aumento no preço do minério de ferro, esta era classificada como positiva, e negativa, caso contrário. Quanto a notícia apresentava uma queda da bolsa de valores, esta foi classificada como negativa, e positiva quando representava uma subida da bolsa. Outra regra definida era com relação à produção da Vale ou do setor. Quando a produção possuía aumento com relação a períodos anteriores, a notícia recebeu classificação positiva. Analogamente, recebeu classificação negativa para produção inferior aos períodos anteriores. A última regra definida foi para o resultado de faturamento quando comparado com anos anteriores. Caso a notícia apresentasse um resultado de faturamento inferior à períodos anteriores, essa era classificada negativamente. Em casos de aumento de faturamento, classificação positiva. Para todas as demais notícias que não se encaixavam nas classificações acima, foram analisadas individualmente quanto à sua polaridade. Foram classificadas 665 notícias positivas e 473 notícias negativas.

Feito isso, foi dada continuidade ao pré-processamento utilizando o *software* WEKA, programa *open source* desenvolvido pela Universidade de Waikato e que possui uma coleção de algoritmos para *machine learning*. Foi selecionado esse *software* pela facilidade de utilização e variedade de funcionalidades disponíveis. O WEKA reconhece somente base de dados do Excel em formato .csv separado por vírgulas. Para evitar erros no pré-processamento do texto, antes de inserir a base de notícias no WEKA, foi necessário retirar todas as vírgulas e ponto e vírgula presentes no texto. Foi percebido também, que o símbolo “%” não era reconhecido pelo programa. Dessa forma, foi feita a alteração do Excel de todos “%” presentes na base de dados para a palavra “percentual”. O mesmo problema acontecia quando estavam presentes aspas, seja ela simples ou dupla. Sem prejuízo para a análise, foram retiradas todas as aspas presentes na base. Finalizados os ajustes, era possível inserir a base de dados no WEKA.

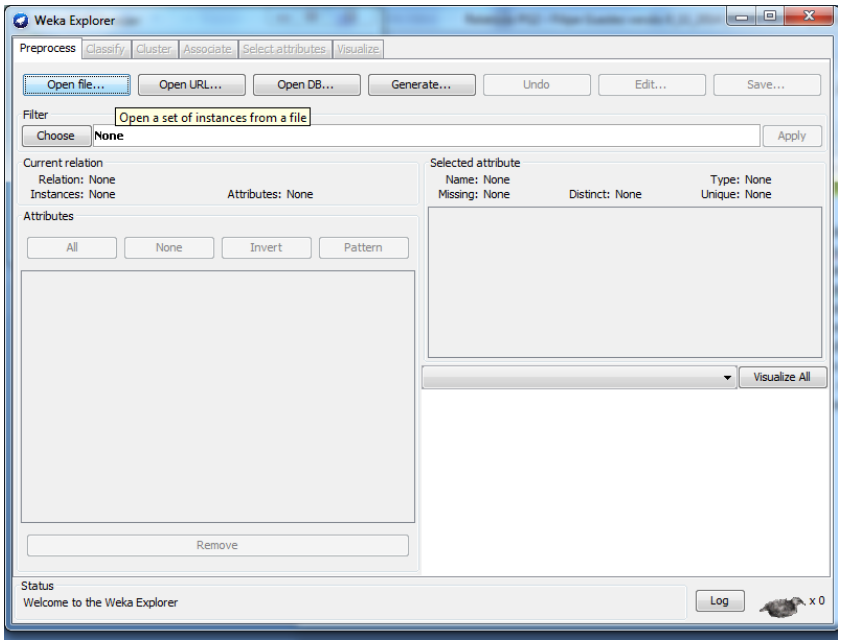

Figura 16 - Importação da base de dados do Excel para o WEKA

Fonte: Elaborado pelo autor

Importada a base em formato .csv para o WEKA, a primeira ação foi convertê-la para .arff, o formato padrão que permite trabalhar no *software*.

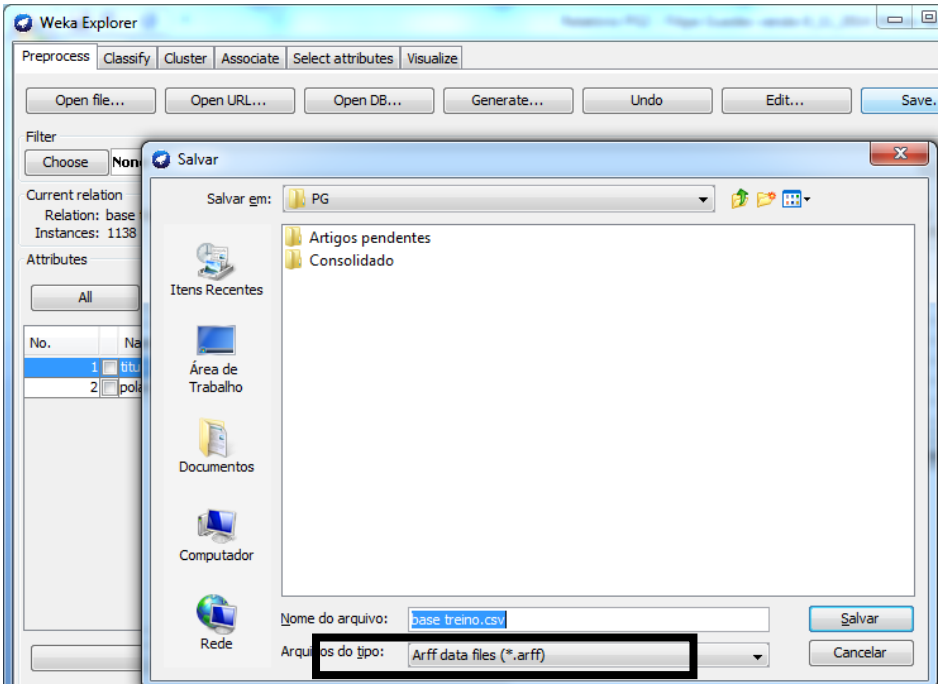

Figura 17 - Conversão do formato .csv para .arrff

Fonte: Elaborado pelo autor

Em seguida, foi definida a coluna de polaridade como classe. Ou seja, foi informado ao software, que aquela coluna representava a classificação da notícia. Essa etapa é apresentada na Figura 18.

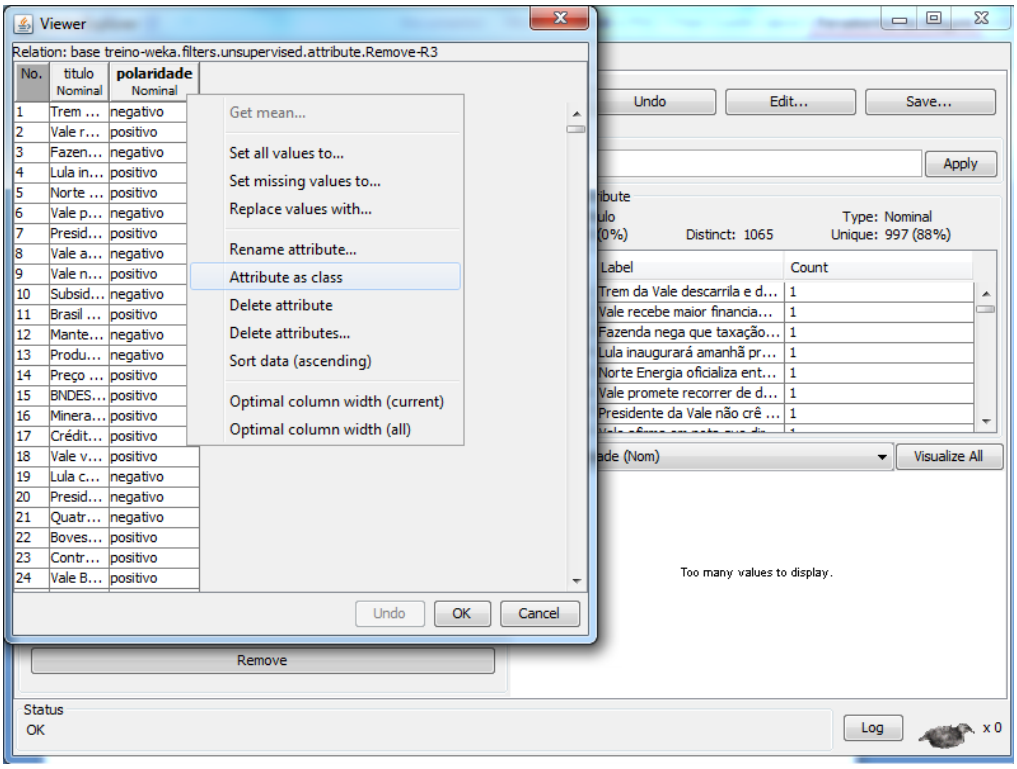

Figura 18 - Definição das classes da base de dados

Fonte: Elaborado pelo autor

O processo de definição das classes dentro da base de dados serve para informar ao algoritmo que irá treinar e testar o classificador, em qual coluna estão as classificações das notícias e quais são as possíveis classes que podem ser atribuídas.

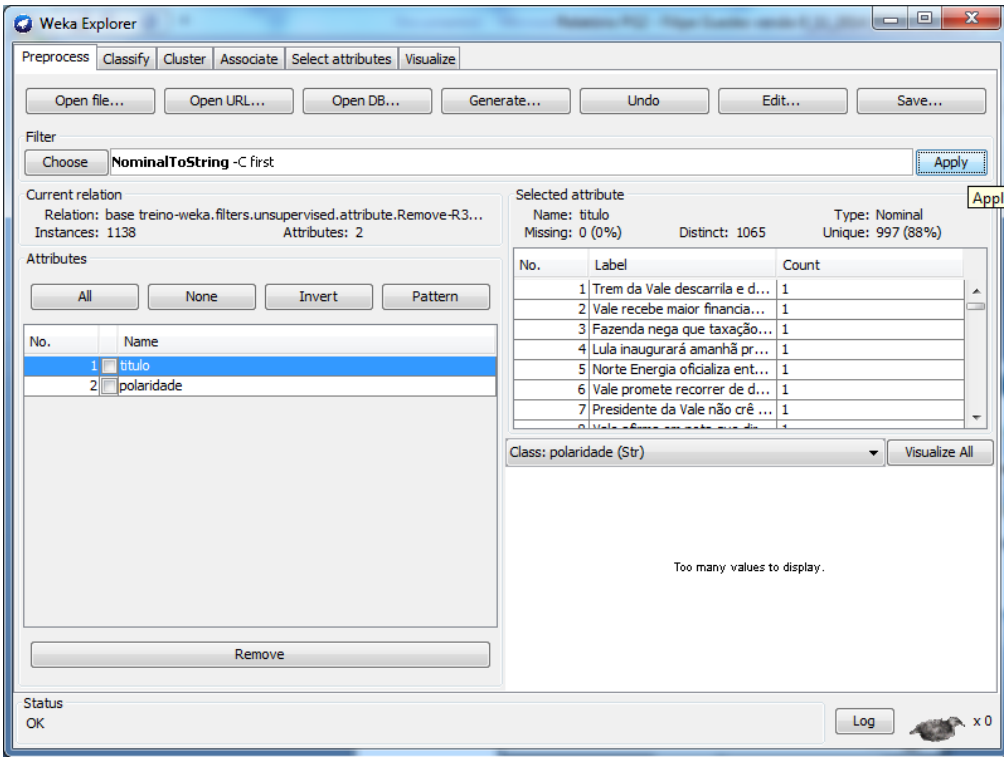

Figura 19 - Transformação da variável título em *string*

Fonte: Elaborado pelo autor

É possível perceber que o título estava em um formato “nominal”. Porém, para realizar a segmentação de palavras, era necessário que a variável estivesse como *string*. Dessa forma, é preciso transformar a variável “título” que estava classificada como nominal em *string*. A transformação foi feita pela função “*NominaltoString*”.

Transformada em *string*, agora era possível fazer a *tokenization*, utilizando a função “*StringToWordVector*”. Após a tokenização finalizada, seu resultado é uma matriz tabela que mensura a presença ou não das palavras. O processo de transformação da variável *string* em vetor pode ser vista na Figura 20.

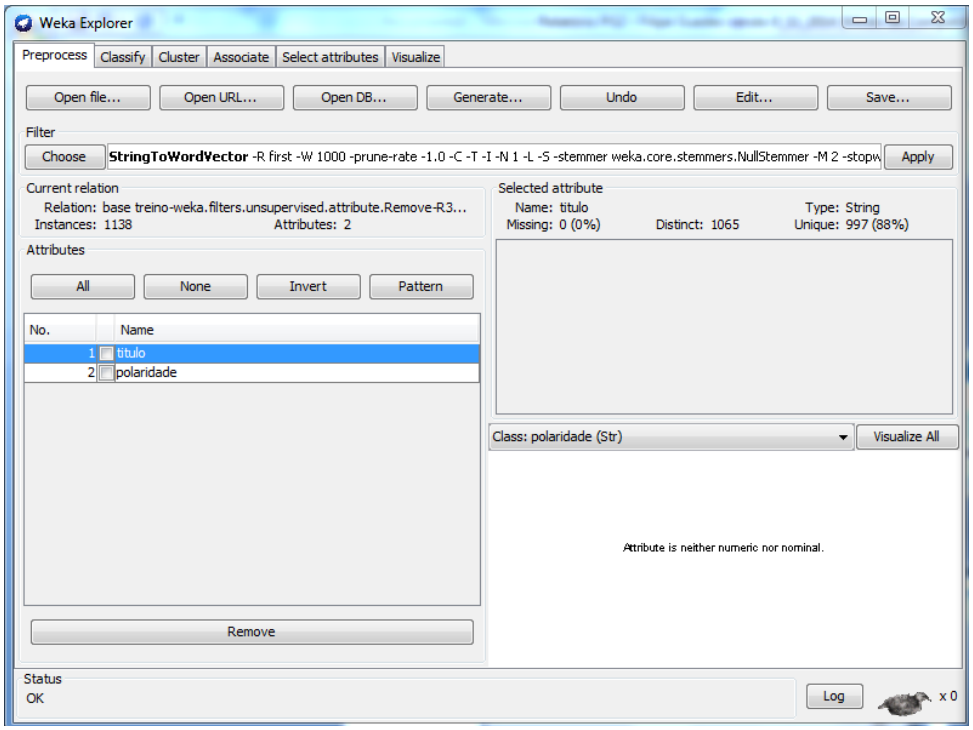

Figura 20 - Seleção da função *"StringToWordVector"*

Fonte: Elaborado pelo autor

O Weka fornece 16 parâmetros para serem definidos durante a etapa de pré-processamento, como apresenta a Figura 21. Durante os primeiros testes, serão variados a maioria desses 16 parâmetros de tokenização, a fim de identificar a melhor parametrização para os dados coletados. Cabe ressaltar que alguns como *attributenameprefix*, *periodicprunning e stemmer*, por exemplo, não serão utilizados, portanto, a variação dos parâmetros não será exaustiva. A justificativa para a não utilização dessas três alternativas será explicada mais a frente.

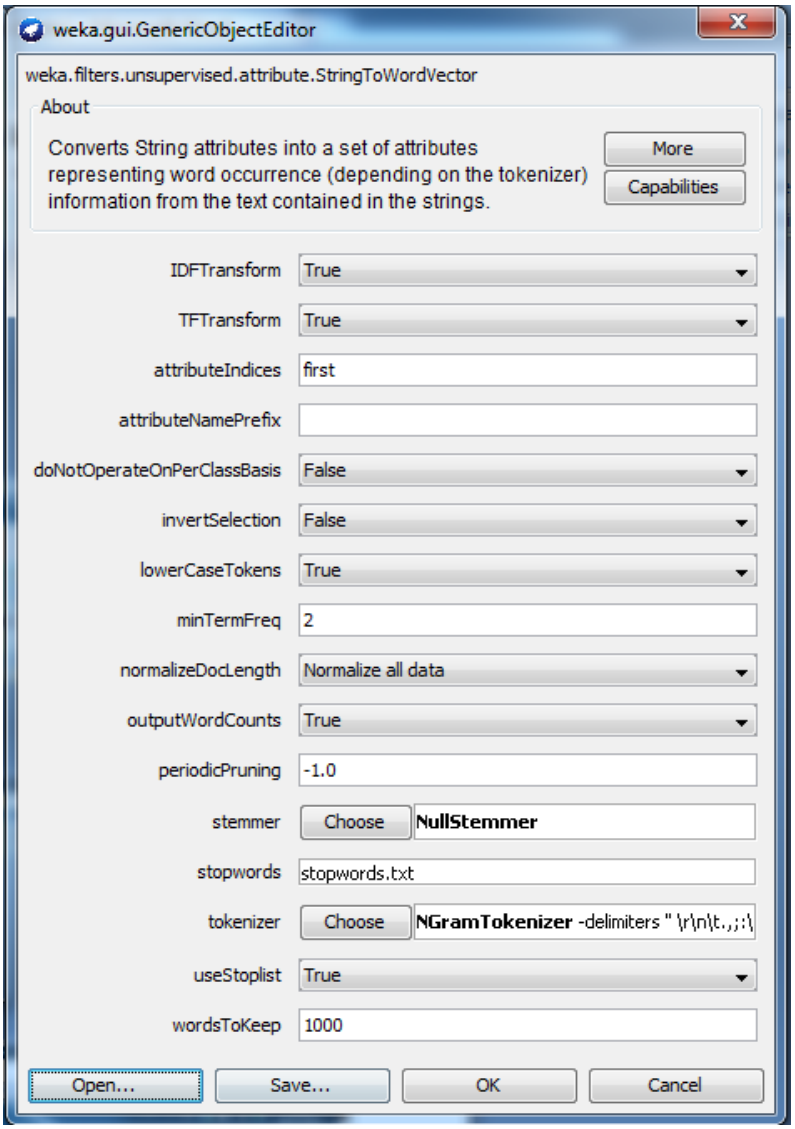

Figura 21 - Definição de parâmetros para realizar a segmentação de palavras
Fonte: Elaborado pelo autor

Os parâmetros da função *StringToWordVector* podem ser visualizados na Figura 21. Essa função consiste em transformar o título da notícia em uma série de atributos representando a ocorrência de cada palavra. Cada título fica representado por um vetor, e esse vetor é determinado pela ocorrência das palavras. O resultado dessa função de pré-processamento é uma matriz onde cada linha será uma notícia e cada coluna uma palavra presente em toda a base. O cruzamento da linha com a coluna diz se a palavra está contida no título da notícia ou não. Um exemplo pode ser visto para a frase “Bovespa inverte tendência e volta a cair com mineração e Petrobras”.

Tabela 3 - Exemplo resultado *tokenization*

Notícia	Bovespa	inverte	Tendência	e	sobe	volta	a	cair	Com	Mineração	Petrobras
1	1	1	1	1	0	1	1	1	1	1	1

Fonte: Elaborado pelo autor

Em casos onde a palavra ocorre na notícia, ela recebe o valor 1, e em casos onde ela não aparece, recebe valor 0. É possível perceber que algumas palavras presentes na frase como "e", "a" e "com", não agregam valor para a análise da polaridade da notícia, pelo contrário, somente atrapalham aumentando o tamanho da matriz. Sabendo isso, é possível retirar essa lista de palavras da matriz para facilitar o processamento e criação do classificador por meio da lista de *stopwords*. Para o presente projeto, a lista de *stopwords* em português foi retirada da base do software R e possui aproximadamente 200 palavras. A relação completa com a lista de *stopwords* está presente no Anexo A desse documento. Ao utilizar uma lista de *stopwords*, deve-se selecionar a opção *True* para a opção *useStoplist* e com isso, as palavras contidas na lista de *stopwords* são ignoradas.

É possível perceber também que sempre a primeira palavra do título vem em letra maiúscula. Essa situação fica mais fácil de ser compreendida com os dois exemplos abaixo

"Julgamento da privatização da Vale não beneficia o país diz **presidente** da mineradora"

"**Presidente** da Vale não crê em novos aumentos de preço do minério de ferro"

Pelo fato da palavra "presidente" em uma notícia estar com letra inicial maiúscula e em outra estar com letra minúscula, isso será interpretado pelos algoritmos, como duas palavras diferentes. Para evitar que uma palavra que aparece com inicial maiúscula e minúscula seja representada duas vezes, é possível reduzir todas as letras para minúscula com a opção *True* para *lowerCaseTokens*. No exemplo apresentado acima, é possível verificar que uma mesma palavra pode aparecer mais de uma vez dentro do mesmo título. Para o resultado da matriz representar a quantidade de vezes que a palavra aparece dentro do texto e não somente uma variável boleana indicando se aparece ou não, basta alterar para *True* o campo *outputWordCounts*.

Para separar o título em palavras, usa-se a opção *tokenizer*, que criará os *tokens*. A vantagem desse processo é transformar o texto em dimensões estruturadas possíveis de se analisar. O WEKA permite três formas de tokenizar o texto. *Word Tokenizer, N-Gram Tokenizer* ou *Alphabetic Tokenizer*. O *Word Tokenizer* é o mais simples de todos e é

conhecido como *bag-of-words*, e consiste em criar um *token* para cada palavra. Alphabetic *Tokenizer* separa os tokens somente com caracteres alfabéticos, ignorando os números. Como no exemplo estudado existem alguns números, não será utilizada essa técnica. A abordagem *N-Gram* cria *tokens* com uma sequência de N palavras. Ou seja, uma tokenização utilizando *2-Gram* criará *tokens* com duas palavras. É possível definir o tamanho mínimo e máximo do *token*. O *N-Gram* é interessante para pegar o contexto do texto, o que não é conseguido utilizando *bag-of-words*.

As demais opções de personalização da estão apresentadas abaixo:

IDFTransform (inverse document frequency transform) e *TFTransform (term frequency transform)* representam a importância da palavra para o conjunto de palavras de acordo com a sua frequência, ou seja, atribuem pesos para os atributos de acordo com sua frequência. O *TFtransform* representa um peso para cada termo a partir da sua frequência. Porém, alguns termos podem aparecer muitas vezes na base, e o *IDFtransform* utiliza o inverso da frequência para diminuir o peso dos termos que aparecem muitas vezes. O segundo método assume a premissa de que os atributos que aparecem poucas vezes são valiosos para a classificação, enquanto o primeiro acredita que os mais freqüentes são mais importantes. Selecionado o *TFTransform* e *IDFTransform* para *true*, o processamento definirá quando é melhor utilizar a frequência dos termos ou o inverso da frequência para determinar a importância da palavra.

attributeIndices – na base estudada, o campo notícias vem na primeira coluna e na segunda coluna tem-se a polaridade da notícia. Como se deseja transformar somente o título da notícia em vetor, nesse campo colocamos a opção *first* para representar que somente a primeira coluna será transformada

attibuteNamePrefix – essa opção é utilizada caso queira inserir algum prefixo aos atributos. Não será utilizada durante as análises

doNotOperateOnPerClassBasis – se selecionada a opção *True*, a opção *minTermFreq* e *wordsToKeep* serão definidos para todo as classes juntas. Selecionada a opção *False*, será definida a *minTermFreq* e *wordsToKeep* para cada classe.

invertSelection – se definido como *false*, somente os atributos selecionados serão trabalhados. Como estamos trabalhando somente com o título, será definido como falso.

normalizeDocLeght – se selecionado como *true*, irá definir se a frequência das palavras deve ser normalizada ou não.

periodicPruning – consiste em definir a taxa de redução da base de palavras selecionadas. Como a base não é muito grande, será definido o valor padrão -1, que significa que não será feita redução.

stemmer – o processo de *stemming* consiste em reduzir as palavras ao seu radical transformando palavras com o mesmo radical em termos semelhantes. Por conta de não ter um arquivo já consolidado para o português, essa função não será utilizada, portanto será selecionada a opção *NullStemmer.*

wordstoKeep – representa a quantidade de palavras a serem utilizadas para definição da polaridade. Esse valor é somente para definição do tamanho aproximado de colunas da matriz. Será utilizado o valor padrão do software de 1000 palavras.

O resultado da segmentação de palavras é a matriz tabela como representa a Figura 22.

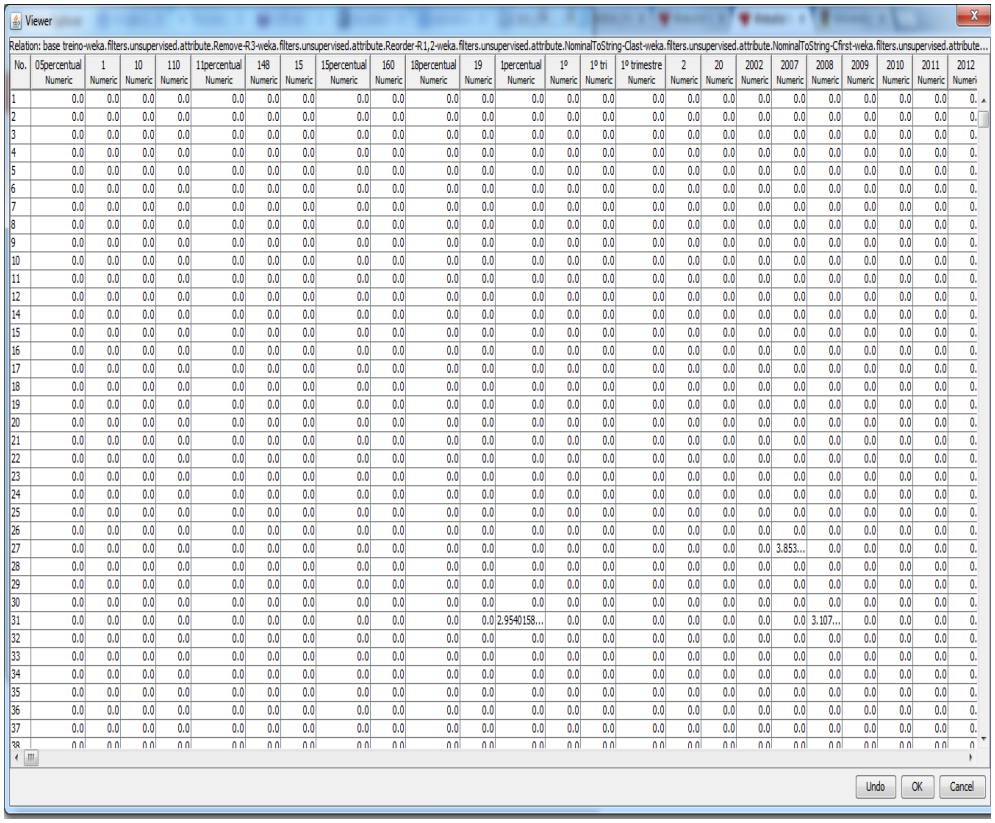

Figura 22 - Resultado da função *StringToWordVector*

Fonte: Elaborado pelo autor

Como explicado anteriormente, onde na matriz tabela apresenta um valor diferente de zero, representa que aquele *token* está presente na notícia da linha y. Exemplificando com base na Figura 22. Na linha 27 é possível visualizar o número 3,853, referente à coluna 2007, isso significa que o número 2007 está presente na notícia 27. Após a *tokenization*, é necessário transformar novamente a polaridade da notícia em classes.

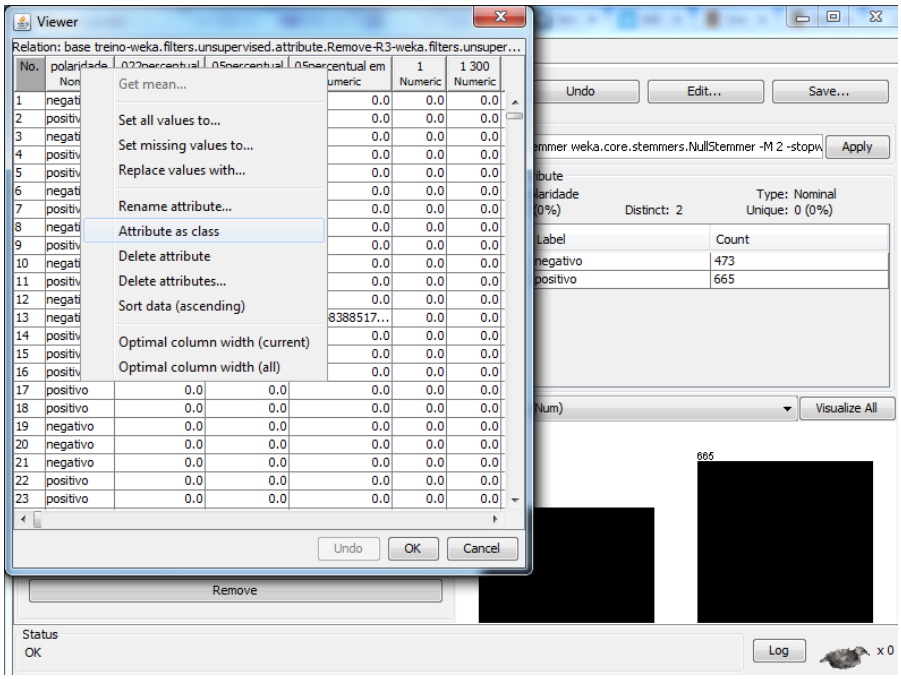

Figura 23 - Transformação da polaridade em classe

Fonte: Elaborado pelo autor

É possível gerar também a lista de palavras mais redundantes para a futura geração do classificador. No exemplo mostrado na Figura 24 é possível ver como a palavra “cai” esta associada às classes. Os pontos azuis representam as vezes que ela está presente nas notícias classificadas como negativas e em vermelho as vezes que ela aparece nas notícias positivas. Percebe-se que ela está muito mais associada à polaridade negativa do que positiva.

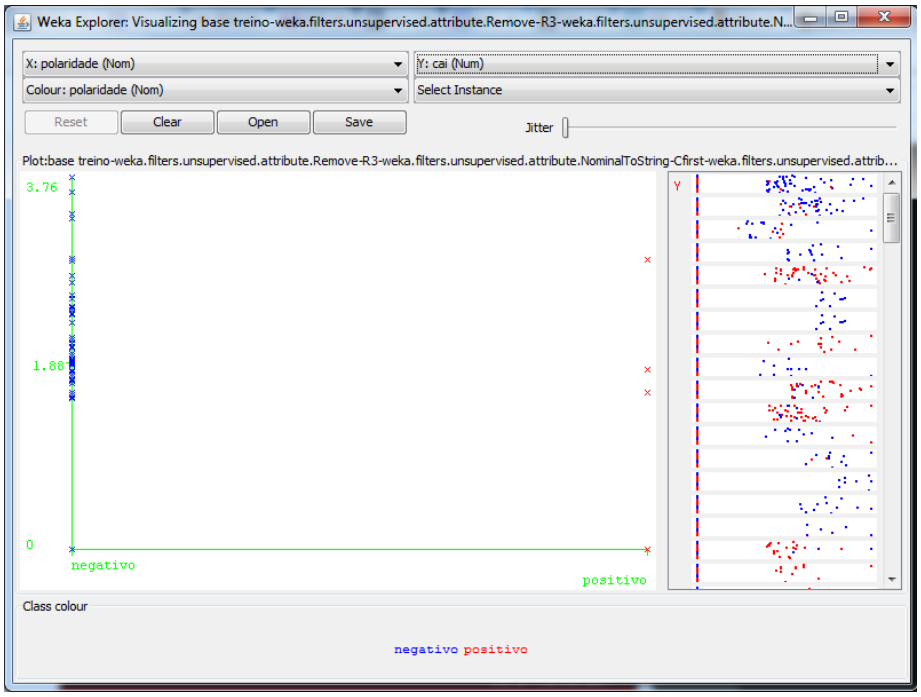

Figura 24 - Exemplo sentimento palavra "cai"

Fonte: Elaborado pelo autor

Na Figura 25, o exemplo apresentado é para a palavra “argentina”. Há um maior equilíbrio entre as classificações positivas e negativas, com ligeira tendência para o sentimento negativo. Como pode-se verificar, para os casos onde a palavra está com associada tanto a sentimentos negativos quanto positivos, haverá maior dificuldade do classificador para definir a tendência de uma notícia que contenha esse termo.

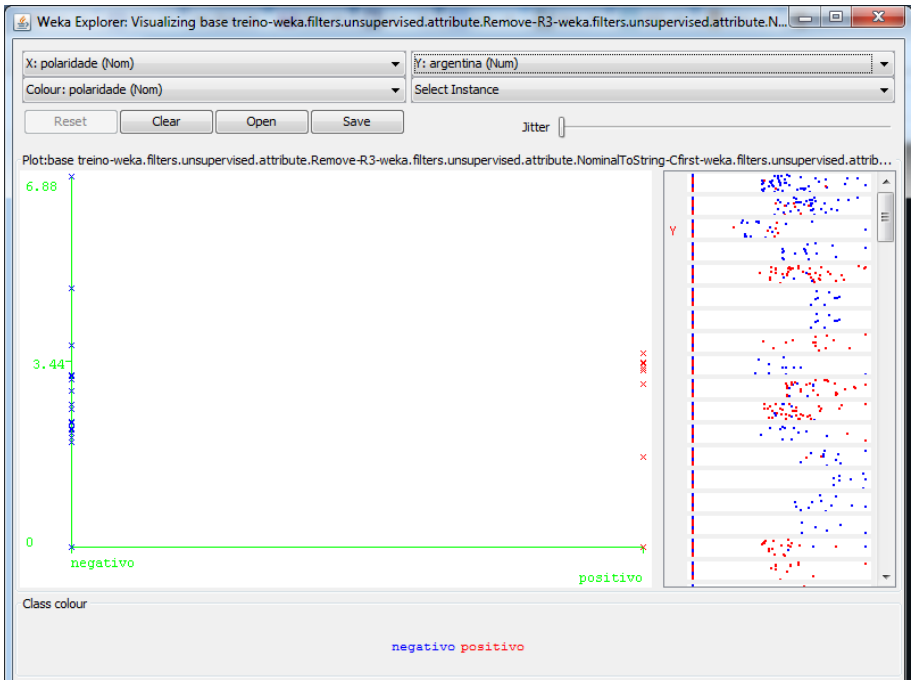

Figura 25 - Exemplo sentimento palavra "argentina"

Fonte: Elaborado pelo autor

A Figura 26 representa o gráfico para a palavra “sobe”. Ao contrário da palavra “argentina”, percebe-se um sentimento positivo muito mais forte, por conta do número de vezes que ela aparece associada à polaridade positiva da notícia. Portanto, em novas notícias onde apareça a palavra “sobe”, existirá uma grande probabilidade de o classificador defini-la como positiva.

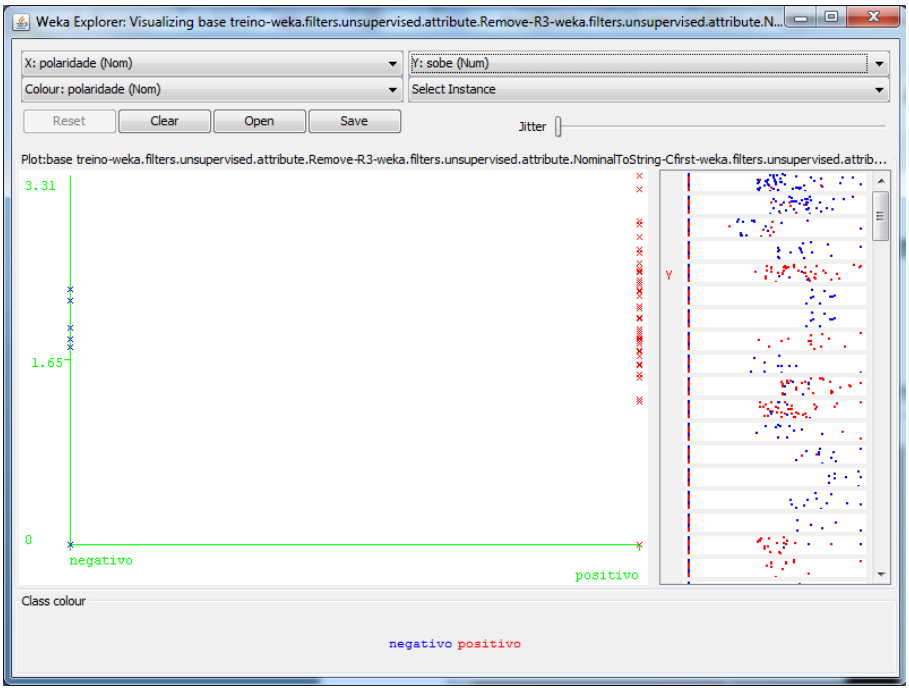

Figura 26 - Exemplo sentimento palavra "sobe"

Fonte: Elaborado pelo autor

Entendidos todos os campos para definição da matriz de *tokens*, foram definidos os parâmetros para o pré-processamento. Foram definidos os parâmetros iniciais para o pré-processamento e em seguida foram treinados e testados os classificadores e seus resultados registrados e analisados. Após a primeira análise, foi alterado um parâmetro a fim de analisar o impacto desse novo parâmetro nos indicadores de Acurácia, *F-Measure*, Área ROC e estatística Kappa. Buscando imparcialidade na análise e comparação dos classificadores, foi feita uma ponderação entre os classificadores para verificar aquele que apresenta os melhores resultados. A ponderação foi realizada da seguinte maneira:

$$Performance = \frac{3 * Acurácia + 2 * F - measure + 2 * Área\ ROC + 3 * Kappa}{10} \quad (18)$$

Como todos os indicadores possuem resultados entre 0 e 1, não se faz necessária uma normalização entre eles. Foi dado peso maior para a acurácia e a estatística Kappa, pelo fato de serem indicadores mais relevantes para a análise da performance quando comparados com o *F-measure* e Área ROC. A atribuição dos pesos a cada um dos indicadores foi feita levando em consideração a revisão de estudos relacionados e a importância que se desejava dar ao indicador global. A acurácia do classificador recebeu maior importância por ser o indicador

mais utilizado na literatura. Dessa maneira, se mostra um indicador muito consistente. Dos estudos analisados, aproximadamente 75% utilizaram a acurácia para analisar o rendimento do classificador. A estatística Kappa, apesar de menos utilizada pelos estudiosos, também recebeu maior peso que os outros indicadores, pois se desejava obter um classificador com grande confiabilidade e aleatoriedade. Dessa forma, como esse indicador era o que melhor representava esse objetivo, recebeu peso levemente superior aos demais.

5.2.1 IDENTIFICAÇÃO DOS MELHORES PARÂMETROS PARA TOKENIZAÇÃO

O primeiro pré-processamento foi realizado utilizando a segmentação por palavras conhecida como *bag-of-words*, ou seja, separação de grupos de uma palavra. Foi utilizada a lista de *stopwords* do R, já mencionada anteriormente. Também foi feita a redução das letras maiúsculas, e definição da frequência mínima dos termos igual a 1, ou seja, mesmo que a palavra apareça somente uma vez, ela estava presente na matriz tabela. Em seguida foram criados os 5 classificadores e o resultado está apresentado na Tabela 4. Para esses parâmetros definidos, o pré-processamento gerou 2409 atributos

Tabela 4 - Resultados classificadores *bag-of-words*

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	76,71	0,767	0,839	0,5190	0,70703
SVM	73,98	0,739	0,728	0,4603	0,65343
Random Forest	73,98	0,733	0,801	0,4470	0,66284
C4.5	68,54	0,663	0,691	0,3094	0,56924
Nearest Neighbor	71,17	0,699	0,720	0,3774	0,61053

Fonte: Elaborado pelo autor

Todos os classificadores foram treinados e testados utilizando a opção 10-*fold cross-validation*. Nesse método, a base de notícias foi dividida em dez partes iguais e o treino e teste ocorreu 10 vezes, sempre utilizando partes diferentes para o teste do modelo. Os resultados apresentados representam a média das dez vezes que o processo rodou. Essa divisão da base em dez partes se manteve fixa para todos os testes seguintes, a menos que explicitado o contrário.

Claramente, o classificador que obteve melhores resultados foi o *Naive Bayes Multinomial*. Para todos os indicadores obteve valores superiores aos demais, enquanto o classificador C4.5 foi o pior classificador para todos os modelos testados. SVM e *Random Forest* obtiveram resultados bem próximos. Foi então alterada a forma de tokenização para confrontar os resultados. Dessa fez foi utilizado a abordagem *n-gram* com n igual a 2 e os

demais parâmetros inalterados. Para esses novos parâmetros, o pré-processamento gerou 6751 atributos. Houve um aumento substancial no número de atributos quando comparado com o pré-processamento anterior. Os resultados estão expostos na Tabela 5.

Tabela 5 - Resultados classificadores 2-*gram*

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	70,65	0,708	0,817	0,4199	0,64292
SVM	74,25	0,736	0,720	0,4535	0,65000
Random Forest	70,91	0,684	0,755	0,3562	0,60739
C4.5	60,36	0,549	0,539	0,1028	0,42952
Nearest Neighbor	68,27	0,647	0,650	0,2895	0,55106

Fonte: Elaborado pelo autor

Com essa nova parametrização, todos os classificadores perderam performance. Somente o SVM obteve valores parecidos com os anteriores, enquanto todos os demais obtiveram rendimentos consideravelmente inferiores. Uma possível justificativa para essa situação é o grande número de atributos presentes, o que aumenta a complexidade do classificador e consequentemente, maior probabilidade de erros. Witten *et al*. (2011) afirmam que uma grande quantidade de atributos geralmente não apresenta boa performance. Liu (2009) afirma que a complexidade do classificador pode causar *overfitting*, que é o fenômeno no qual o algoritmo de *machine learning* não consegue generalizar os dados de maneira adequada. Outra justificativa para a perda de performance é explicada por Hagenau *et al*. (2013) que mostraram a partir de estudos, que 3-*gram* apresenta um número elevado de combinações, o que reduz o número de frequência de cada atributo e consequentemente, reduz a eficiência do classificador. Essa justificativa parece mais pertinente para a situação estudada. Como há uma quantidade de atributos seis vezes maior que o número de notícias, a grande maioria dos atributos aparece uma só vez, o que aumenta a probabilidade de erro do classificador.

Foram definidos então novos parâmetros para o pré-processamento. Mais uma vez foi alterado somente as informações da tokenização. Foi aumentado o n do *n-gram* de 2 para 3. A hipótese era que mais uma vez, o número de atributos aumentaria e a performance dos classificadores seria menor. A primeira hipótese foi confirmada. Foram gerados 8069 atributos, aumento de aproximadamente 20% no número de atributos para criação dos classificadores. Os resultados estão mostrados na Tabela 6.

Tabela 6 - Resultados classificadores 3-gram

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	62,91	0,618	0,765	0,3034	0,55635
SVM	71,26	0,689	0,670	0,3656	0,59526
Random Forest	65,55	0,586	0,695	0,2034	0,51387
C4.5	58,61	0,435	0,508	0,0049	0,36590
Nearest Neighbor	66,34	0,598	0,576	0,2230	0,50072

Fonte: Elaborado pelo autor

A segunda hipótese também foi confirmada. Houve grande queda de performance de todos os classificadores, inclusive do SVM. Uma informação que chama atenção é referente ao C4.5. Mais uma vez ele teve todos os indicadores com valor inferior aos demais. A acurácia foi somente de 58,61, pouco acima de uma classificação aleatória. A estatística Kappa também teve valor próximo de 0, o que mostra a total aleatoriedade do modelo, ou seja, é um classificador sem nenhuma confiabilidade e precisão para o caso estudado.

Foi feito então, mais uma variação na tokenização. O valor n do *n-gram* foi alterado mais uma vez, agora variando entre 1 e 2. Ou seja, as palavras foram separadas em grupos de 1 ou 2 palavras. Foram obtidos 2315 atributos e os resultados estão representados na Tabela 7.

Tabela 7 - Resultados classificadores 1,2-gram

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	80,58	0,806	0,890	0,6032	0,76190
SVM	78,03	0,779	0,769	0,5433	0,70401
Random Forest	74,78	0,749	0,817	0,4841	0,68277
C4.5	68,27	0,669	0,694	0,3656	0,58709
Nearest Neighbor	67,31	0,675	0,706	0,3550	0,58463

Fonte: Elaborado pelo autor

Os resultados mostram que a performance foi melhor que todos os anteriores, com exceção do KNN. Para os demais, houve ganho considerável tanto de acurácia, quanto de *F-measure*, ROC área, e estatística Kappa, o que mostra que essa nova parametrização gerou classificadores mais eficientes, mais confiáveis, mais precisos e com menor índice de falsos positivos.

A próxima mudança continua no tamanho do n do *n-gram.* Agora o intervalo foi ampliado para n entre um e três. Para essa situação, foram gerados 3288 atributos.Os resultados obtidos foram apresentados na Tabela 8.

Tabela 8 - Resultados classificadores 1, 2 e 3-gram

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	80,58	0,807	0,889	0,6040	0,76214
SVM	77,50	0,774	0,764	0,5323	0,69979
Random Forest	74,69	0,748	0,818	0,4831	0,68220
C4.5	67,83	0,661	0,686	0,3012	0,56325
Nearest Neighbor	66,78	0,670	0,707	0,3458	0,57948

Fonte: Elaborado pelo autor

Os resultados da performance para 1,2 e 3-*gram* foram bastante parecidos com a segmentação de palavras para 1 e 2-*gram*, com uma vantagem pouco considerável para o 1 e 2-*gram*. Como as performances foram muito próximas, foi realizada uma análise mais detalhada dos dados. A acurácia foi ligeiramente maior em todos os classificadores utilizando 1 e 2-*gram*. A área ROC também ficou levemente superior para todos os classificadores com 1 e 2-gram, logo esses modelos representam melhor a realidade. Por conta desses dois indicadores que se destacaram mais, é possível concluir que a segmentação com 1 e 2-*gram* foi superior ao 1,2 e 3-*gram*.

Buscando realizar mais uma comparação, foi alterado mais uma vez o n, agora para 2 e 3-*gram*. Como nas análises anteriores com 2-*gram* e 3-*gram* obteve-se resultados inferiores, a hipótese para esse novo modelo é de que a performance seria inferior às demais. Foram gerados 2334 atributos e os resultados estão representados na Tabela 9.

Tabela 9 - Resultados classificadores 2 e 3-gram

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	78,91	0,789	0,867	0,5669	0,73800
SVM	77,24	0,769	0,754	0,5202	0,69238
Random Forest	75,21	0,748	0,803	0,4781	0,67926
C4.5	60,19	0,551	0,534	0,1019	0,42814
Nearest Neighbor	70,82	0,700	0,743	0,3778	0,61440

Fonte: Elaborado pelo autor

Essa parametrização se mostrou mais eficiente para o *K-Nearest Neighbor*, que obteve sua melhor performance. Em contrapartida, *Naive Bayes multinomial* e C4.5 tiveram perda de performance considerável. Mais uma vez, a estatística Kappa foi muito baixa para o C4.5 o que confirma a baixa robustez desse modelo para a situação em análise. E confirmando as tendências anteriores, o modelo *Naive Bayes Multinomial* apresentou melhor performance quando comparado aos demais.

É possível perceber com as alterações dos parâmetros de tokenização, que os classificadores utilizando *Naive Bayes Multinomial* e *Support Vector Machine* apresentaram

resultados superiores aos demais. E os C4.5 e *nearest neighbor* apresentaram resultados inferiores. Em geral, com a utilização simultânea de 1 e 2-*gram*, obteve-se melhores resultados. Conforme já apresentado por Hagenau *et al*. (2013), 3-*gram* apresenta um número elevado de combinações, o que reduz o número de frequência de cada atributo e consequentemente, reduz a eficiência do classificador. Essa informação pode ser confirmada com a apresentação dos dados. É possível perceber que em média, a acurácia do classificador caiu 5% quando comparado o tokenizador utilizando 2-*gram* e 3-*gram*.

5.2.2 IDENTIFICAÇÃO DOS MELHORES PARÂMETROS PARA FREQUÊNCIA MÍNIMA DOS TERMOS

Como com 1 e 2-*gram* obteve-se os melhores resultados, foi definido esse parâmetro como padrão para os testes seguintes. Nessa etapa foi alterada a frequência mínima que o termo deve aparecer para ser utilizado como base para o classificador. Nos testes anteriores foi definida como frequência mínima igual a 1. Nesses novos testes foram alterados para 3, 4, 5 e 6 vezes. A hipótese é que selecionando somente os atributos que aparecem com maior frequência, se obtenha ganho de performance dos classificadores. O ganho de performance seria obtido pela redução de palavras que aparecem com pouca frequência e que poderiam estar provocando ruídos nos classificadores. Foi definido então o separador de palavras utilizando n-*gram* com n variando entre 1 e 2. Foi mantida a lista de *stopwords* e redução de caracteres maiúsculos. Ou seja, foram mantidos os parâmetros anteriores e alterado a frequência mínima que os termos deveriam aparecer na amostra para serem selecionados como atributos. Quando por exemplo, definido o valor três, todos os termos que aparecem somente uma ou duas vezes não serão definidos como atributo.

Para o pré-processamento com frequência mínima igual a três, foram obtidos 1053 atributos e os resultados estão apresentados na Tabela 10.

Tabela 10 - Resultados classificadores 1 e 2-gram e frequencia mínima = 3

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	78,11	0,782	0,869	0,5524	0,73025
SVM	75,04	0,749	0,740	0,4828	0,66776
Random Forest	73,63	0,736	0,811	0,4570	0,66739
C4.5	68,45	0,669	0,700	0,3170	0,57425
Nearest Neighbor	66,25	0,665	0,690	0,3183	0,56524

Fonte: Elaborado pelo autor

Apesar do ganho de acurácia para o C4.5 e o KNN, para os demais modelos e os outros indicadores, houve perda de resultado. Portanto, a hipótese inicial de que a seleção de atributos que aparecem com maior frequência resultaria em melhores classificadores, a princípio foi rejeitada. A fim de verificar se realmente o aumento da frequência mínima não resulta em melhores classificadores, foi aumentada a frequência mínima para 4 vezes. De 1053 atributos gerados na etapa anterior, agora restaram somente 617. Os resultados estão expostos na Tabela 11.

Tabela 11 - Resultados classificadores 1 e 2-gram e frequencia mínima = 4

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	76,44	0,765	0,847	0,5170	0,70682
SVM	74,25	0,741	0,729	0,4639	0,65592
Random Forest	73,81	0,739	0,810	0,4632	0,67019
C4.5	68,27	0,667	0,697	0,3132	0,57157
Nearest Neighbor	68,80	0,690	0,716	0,3654	0,59722

Fonte: Elaborado pelo autor

Com exceção do *Random Forest* e do *Nearest Neighbor* que obtiveram melhores resultados quando comparados com a frequência mínima igual a dois, os demais tiveram perdas. Mais uma vez a hipótese foi rejeitada. Obtendo as médias de todos os indicadores para a frequência mínima igual a 2 e 3, é possível perceber que em termos de resultados esses classificadores são muito semelhantes, conforme representa a Tabela 12.

Tabela 12 - Comparação entre a média dos resultados entre frequência mínima igual a 2 e 3

	Acurácia	F-Measure	ROC Area	Kappa	Performance
Média Frequencia 2	72,296	0,7202	0,7620	0,42550	0,640978
Média Frequencia 3	72,314	0,7204	0,7598	0,42454	0,640344
Δ	0,0180	0,0002	-0,0022	0,00096	-0,000634

Fonte: Elaborado pelo autor

Portanto, apesar de para alguns modelos ter-se obtido melhores resultados com o aumento da frequência mínima, para outros houve perda, e na média as perdas se igualaram aos ganhos. *Naive Bayes Multinomial* foi o que apresentou maiores perdas, enquanto *K-Nearest Neighbor* foi o que apresentou maiores ganhos. Uma justificativa para o fato da seleção dos atributos mais presentes na base de dados não ter retornado em uma melhoria de performance diz respeito ao fato de que nem sempre os atributos que estão presentes em maior quantidade são os mais representativos e redundantes para a análise. Em alguns casos, um atributo que aparece muitas vezes é menos redundante que outro que aparece somente uma vez, pois ele pode estar presente em ambas as categorias, além do fato de que ele estar

presente muitos documentos, como mostra Liu (2009) acaba por perder importância e característica.

Porém, conforme planejado, se dará continuidade ao aumento da frequência mínima dos termos para 5 e 6. Quando a frequência foi igual a 5, foram obtidos 424 atributos, e para frequência 6, 310 atributos. É possível perceber a grande redução do número de atributos. Inicialmente 2315, foram reduzidos a 1053, depois a 617 e por fim, para 424 e 310 atributos. Os resultados para a frequência mínima igual a 5 e 6 estão resumidos na Tabela 13.

Tabela 13 - Resultados classificadores 1 e 2-gram e frequencia mínima = 5 e 6

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Frequência mínima = 5					
Naive Bayes Multinomial	74,51	0,745	0,829	0,4741	0,68056
SVM	75,13	0,749	0,736	0,4800	0,66639
Random Forest	74,16	0,743	0,819	0,4724	0,67660
C4.5	68,36	0,669	0,701	0,3158	0,57382
Nearest Neighbor	69,77	0,698	0,740	0,3811	0,61124
Frequência mínima = 6					
Naive Bayes Multinomial	73,50	0,734	0,818	0,4513	0,66629
SVM	74,60	0,743	0,731	0,4686	0,65918
Random Forest	71,52	0,715	0,803	0,4132	0,64212
C4.5	68,98	0,677	0,709	0,3323	0,58383
Nearest Neighbor	68,98	0,691	0,746	0,3663	0,60423

Fonte: Elaborado pelo autor

Em ambos os casos houve perda de performance, com algumas exceções pontuais. Agora é possível rejeitar por completo a hipótese de que a seleção de atributos baseada na frequência mínima dos termos geraria melhoria de performance. Em nenhum caso houve ganho considerável, e é possível perceber que a medida que se restringe a frequência mínima, aumenta-se a perda de performance. Além das justificativas já citadas anteriormente para a perda de rendimento, esse tipo de seleção acaba por eliminar muitos atributos relevantes para o classificador. Isso fica claro com a redução drástica do número de atributos, que inicialmente era de 2315 e no último teste, restaram apenas 310, ou seja, redução de mais de 80% dos atributos iniciais. Porém, cabe ressaltar, que a redução drástica pode ser benéfica, porém se fazem necessários algoritmos mais robustos para fazer essa seleção e redução de atributos.

Percebido que a utilização da frequência não se mostrou adequada e somada à explicação de Liu (2009), foi utilizado as medidas TF e TF-IDF durante o pré-processamento para definir quando utilizar a frequência do termo ou seu inverso. A hipótese é de que esses

novos parâmetros gerem melhoria de performance. Conforme já explicado na seção 5.2, essas duas transformações atribuem pesos para cada atributo a partir de sua frequência. E em casos onde a frequência é muito alta, a fim de não enviesar o classificador, o algoritmo utiliza o inverso da frequência para reduzir o peso desses atributos. Portanto, para o próximo teste foi mantido a tokenização utilizando 1 e 2-*gram*, redução de *stopwords* e caracteres maiúsculos e frequência mínima dos termos igual a um. Foram incluídos os termos de frequência (TF) e seu inverso (IDF), quando necessário. Os resultados estão expressos na Tabela 14.

Tabela 14 - Resultados classificadores 1 e 2-gram e TF e IDF

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	82,51	0,826	0,912	0,6460	0,78893
SVM	78,03	0,779	0,769	0,5433	0,70668
Random Forest	74,25	0,743	0,811	0,4737	0,67566
C4.5	68,27	0,669	0,694	0,3150	0,57191
Nearest Neighbor	67,31	0,675	0,706	0,3550	0,58463

Fonte: Elaborado pelo autor

Dentre os classificadores, SVM, C4.5 e KNN não apresentaram nenhuma mudança, logo é possível concluir que para o caso em estudo, o TF e IDF não se mostrou eficaz. Acredita-se que se tivesse uma base maior, os algoritmos seriam mais sensíveis para essas duas tranformações. O classificador *Naive Bayes Multinomial* obteve ganhos de performance em todos os indicadores enquanto o *Random Forest* teve leve perda de performance. Como não é possível afirmar que esses dois parâmetros geram ganhos para o classificador, eles não foram utilizados nos próximos classificadores.

5.2.3 IDENTIFICAÇÃO DOS MELHORES PARÂMETROS PARA SELEÇÃO DE ATRIBUTOS

Os testes seguintes foram utilizados para definir qual método de seleção de atributos gera melhores resultados. A seleção de atributos foi feita utilizando CSF e qui-quadrado na base de notícias e depois treinados e testados os cinco algoritmos de *machine learning*. A hipótese a ser validade nesses casos é de que muitos atributos são irrelevantes e portanto, a quantidade de atributos será reduzida significativamente. Feldman e Sanger (2007) mostram que métodos mais agressivos de seleção de atributos podem reduzir até 99% dos atributos iniciais. A primeira seleção foi realizada com o algoritmo CSF. Buscando manter o poder de comparação, foram mantidos os parâmetros que até o momento tinham obtido os melhores resultados. Foi utilizado a lista de *stopwords*, tokenização com *1 e 2-gram*, remoção de

caracteres maiúsculos e frequência mínima dos termos igual a 1. Os 2315 atributos iniciais foram reduzidos a 137. Os resultados são apresentados na tabela 15.

Tabela 15 - Resultados classificadores com seleção de atributos utilizando CSF

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	84,97	0,844	0,905	0,6763	0,80760
SVM	83,47	0,828	0,897	0,6468	0,78945
Random Forest	83,21	0,825	0,895	0,6376	0,78491
C4.5	66,87	0,607	0,589	0,2372	0,51097
Nearest Neighbor	82,95	0,822	0,892	0,6315	0,78110

Fonte: Elaborado pelo autor

É possível verificar que o classificador KNN obteve performance muito superior à anterior. A acurácia que tinha sido 67,31%, agora ficou em 82,95%. A estatística Kappa, que estava em 0,355, com a seleção de atributos ficou 0,6315. O indicador de performance aumentou de 0,58 para 0,78. Em menor proporção, o classificador *Random Forest* também obteve ganhos consideráveis, assim como SVM e *Naive Bayes Multimomial*. A exceção foi o C4.5, que obteve rendimento inferior com a seleção. Isso mostra que na relação de atributos inicias haviam muitos atributos irrelevantes e redundantes que estavam dificultando o classificador.

Buscando identificar se o método de seleção de atributos utilizando qui-quadrado performa melhor que o CSF foram realizados testes variando o número de atributos finais. Tendo em vista que o qui-quadrado cria um ranking dos melhores atributos, para realizar a seleção dos mesmos é necessário definir quantos atributos se deseja manter. Foi definido como o primeiro valor de atributos igual a 200, ou seja, os 200 melhores atributos de acordo com a estatística qui-quadrado serão mantidos. Os resultados estão na tabela 16.

Tabela 16 - Resultados classificadores com seleção de atributos utilizando qui-quadrado com 200 atributos

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
Naive Bayes Multinomial	82,86	0,821	0,887	0,6297	0,77909
SVM	81,82	0,809	0,788	0,6062	0,74672
Random Forest	80,93	0,799	0,878	0,5850	0,75369
C4.5	66,87	0,607	0,618	0,2372	0,51677
Nearest Neighbor	79,70	0,784	0,860	0,5566	0,73488

Fonte: Elaborado pelo autor

Mesmo com a alteração da forma de seleção dos atributos, mais uma vez o C4.5 não melhorou sua performance. De posse de todos os testes já realizados, é possível concluir antecipadamente que esse é o classificador que obteve piores resultados. Esse resultado já era

esperado, levando em consideração a simplicidade desse classificador quando comparado com os demais. Os quatro outros classificadores obtiveram bons resultados quando confrontados com os resultados obtidos sem a seleção de atributos, porém, obtiveram performance pior que os classificadores gerados com a seleção de atributos utilizando a correlação. Com o intuito de identificar a quantidade de atributos que maximiza a performance dos classificadores, foram realizadas outras variações do numero de atributos a serem mantidos. Foi feito o teste com 250, 300, 350, 400 e 450 atributos. Os resultados estão apresentados na tabela 17.

Tabela 17 - Resultados classificadores com seleção de atributos utilizando qui-quadrado com 250, 300, 350, 400 e 450 atributos

Classificador	Acurácia	F-Measure	ROC Area	Kappa	Performance
250 atributos					
Naive Bayes Multinomial	85,06	0,845	0,908	0,6792	0,80954
SVM	83,21	0,825	0,805	0,638	0,76703
Random Forest	82,60	0,818	0,897	0,6242	0,77806
C4.5	66,87	0,607	0,618	0,2372	0,51677
Nearest Neighbor	80,93	0,799	0,873	0,5861	0,75302
300 atributos					
Naive Bayes Multinomial	85,14	0,846	0,921	0,6818	0,81336
SVM	83,47	0,828	0,808	0,6441	0,77084
Random Forest	82,60	0,818	0,908	0,6246	0,78038
C4.5	66,87	0,607	0,618	0,2372	0,51677
Nearest Neighbor	81,28	0,804	0,880	0,5945	0,75899
350 atributos					
Naive Bayes Multinomial	85,41	0,850	0,924	0,6897	0,81794
SVM	82,95	0,823	0,803	0,6334	0,76407
Random Forest	82,86	0,822	0,903	0,6314	0,78300
C4.5	66,87	0,607	0,618	0,2372	0,51677
Nearest Neighbor	81,19	0,803	0,868	0,5938	0,75591
400 atributos					
Naive Bayes Multinomial	85,67	0,853	0,922	0,6960	0,82081
SVM	82,60	0,820	0,802	0,6275	0,76045
Random Forest	82,33	0,819	0,890	0,6244	0,77611
C4.5	66,87	0,608	0,623	0,2378	0,51815
Nearest Neighbor	79,26	0,785	0,856	0,5543	0,73227
450 atributos					
Naive Bayes Multinomial	84,07	0,847	0,916	0,6826	0,80959
SVM	81,45	0,809	0,790	0,6034	0,74517
Random Forest	81,45	0,811	0,878	0,6081	0,76458
C4.5	67,39	0,616	0,631	0,2514	0,52699

Nearest Neighbor	77,68	0,769	0,832	0,5225	0,70999

Fonte: Elaborado pelo autor

É possível afirmar que levando em consideração os cinco classificadores, os melhores resultados foram obtidos com 300 atributos. Quando foi aumentado para 350 e 400 atributos, somente o classificador *Naive Bayes Multinomial* obteve aumento de performance, enquanto os demais começaram a cair. Quando aumentada a seleção de atributos para 450, todos obtiveram perda de rendimento. Considerando os resultados obtidos com a seleção de atributos com qui-quadrado com 300 atributos e comparando com o CSF e ignorando os dados do classificador C4.5 que obteve performance muito inferior aos demais, é possível perceber que somente para o classificador *Naive Bayes Multinomial*, o qui-quadrado obteve melhor performance, enquanto o CSF obteve melhor rendimento para o *K-nearest neighbor*, SVM e *Random Forest*. Em geral o CSF leva vantagem para o qui-quadrado por conta da sua complexidade. Enquanto o qui-quadrado analisa a relação de cada atributo separadamente com a sua classe, o CSF analisa a correlação entre atributos e a correlação dos atributos com as classes. Assim, consegue melhor eliminar os atributos redundantes que o qui-quadrado.

5.2.4 IDENTIFICAÇÃO DAS PALAVRAS COM MAIOR IMPACTO NA POLARIDADE DAS NOTÍCIAS

A identificação das palavras com maior impacto na polaridade foi feita com base nos atributos remanescentes da seleção. Como o filtro foi feito utilizando a função "*Ranker*", os primeiros atributos na lista são os que melhor classificam a notícia.

Como pode ser verificado na Figura 27 o software WEKA fornece a lista dos atributos ranqueados de acordo com seu poder de definição de polaridade.

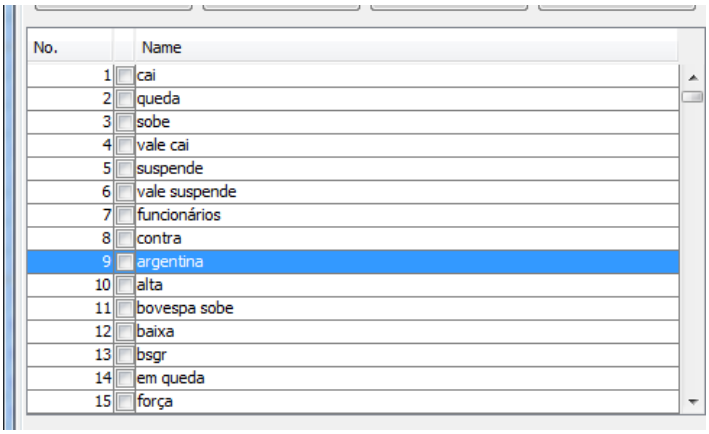

Figura 27 - Atributos mais relevantes (qui-quadrado)
Fonte: Elaborado pelo autor

A Figura 27 mostra os 15 atributos mais relevantes a partir da classificação utilizando a estatística qui-quadrado já explicada anteriormente na seção 2.4.5. Como a segmentação de palavras foi feita utilizando n-*gram* com tamanhos 1 e 2, os atributos são compostos por grupos de um ou duas palavras. Porém, essa lista não permite visualizar se o sentimento associado às palavras é positivo ou negativo. O software também permite identificar o sentimento associado a cada palavra por meio de gráficos. A Figura 28 apresenta as informações para as três principais palavras.

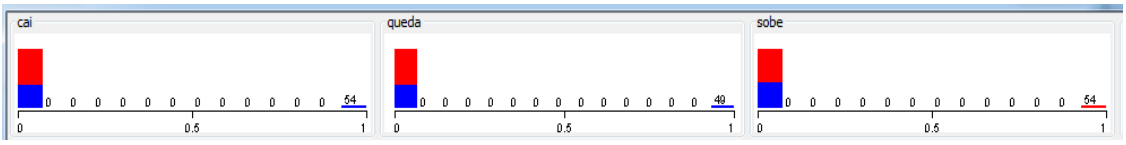

Figura 28 - Sentimento associado às três palavras mais relevantes (qui-quadrado)
Fonte: Elaborado pelo autor

A imagem deve ser entendida da seguinte maneira. A cor azul está associada ao sentimento negativo, enquanto a cor vermelha está associada ao sentimento positivo. A coluna associada ao valor 0 representa o número de documentos onde a palavra ou grupo de palavras não foi encontrado. E a coluna associada ao valor 1 representa o total de documentos onde a palavra foi encontrada. Associando essas duas informações, é possível perceber, que a palavra "cai" esteve presente em 54 documentos classificados como negativo, assim como a palavra queda, que esteve presente em 49 documentos negativos. Por sua vez, a palavra sobe esteve presente em 54 documentos classificados como positivos. Seguindo a análise, a lista das10 palavras mais relevantes para cada classe a partir da estatística qui-quadrado, estão sintetizadas na tabela 18.

Tabela 18 - 10 palavras mais relevantes para classificação, por sentimento (qui-quadrado)

Ranking	Positivo	Negativo
1	Sobe	Cai
2	Alta	Queda
3	Bovespa sobe	vale cai
4	BNDES	suspende
5	venda de	vale suspende
6	vale sobe	funcionários
7	anuncia	Contra
8	obtém licença	Argentina
9	vale obtém	Baixa
10	Obtém	Bsgr

Fonte: Elaborado pelo autor

Foi realizado o mesmo trabalho para os atributos selecionados pelo CSF. Os 15 atributos mais relevantes estão representados na Figura 29. Comparando os dois resultados contendo os 15 principais grupos de palavras, houve pouca mudança. Os atributos são os mesmos. A única diferença é a ordem da quinta e sexta palavras que foram alteradas de um para outro.

Figura 29 - Atributos mais relevantes (CSF)
Fonte: Elaborado pelo autor

A lista das 10 palavras mais relevantes para cada classe de acordo com a seleção de atributos por CSF também pouco difere da lista anterior. A lista de palavras associadas ao sentimento negativo não mudou, alterando somente a posição de duas palavras. Já a lista para o sentimento positivo teve alteração de uma palavra. As 10 palavras mais relevantes para cada classificação estão apresentadas na tabela 19.

Tabela 19 - 10 palavras mais relevantes para classificação, por sentimento (CSF)

Ranking	Positivo	Negativo
1	Sobe	Cai
2	Alta	Queda
3	Bovespa sobe	vale cai
4	BNDES	vale suspende
5	venda de	Suspende
6	vale sobe	funcionários
7	Anuncia	Contra
8	Obtém	Argentina
9	cade aprova	Baixa
10	obtém licença	Bsgr

Fonte: Elaborado pelo autor

Um fato que chamou a atenção foi a grande quantidade de notícias negativas nas primeiras posições do ranking geral. Das dez palavras mais bem ranqueadas, oito estavam

associadas ao sentimento negativo, enquanto somente duas estavam associadas ao negativo. Para se obter as dez palavras que mais representam o sentimento negativo, para as duas situações foi conseguido com as treze primeiras palavras, enquanto as dez palavras associadas à polaridade positiva só foi conseguida com a trigésima terceira palavra para o CSF e trigésima sétima para o qui-quadrado. Essa grande quantidade de palavras negativas nas primeiras posições pode representar uma maior facilidade do classificador julgar uma notícia como negativa, o que pode indicar um viés.

Outro fato que chama a atenção é a presença das palavras Argentina e bsgr na lista de termos com sentimento negativo. O sentimento negativo associado à Argentina é explicado por uma série de conflitos entre a Vale e o governo argentino em março de 2013 que podem ser exemplificados com as notícias retiradas da base. Em 12 de março de 2013 o jornal Estadão publicou a seguinte notícia:

"Argentina diz que tomará medidas por projeto na Vale"

No dia 13 de março de 2013 o mesmo jornal publicou:

"Analistas criticam pressão da Argentina sobre a Vale"

No dia seguinte, 14 de março, a Agência Brasil divulgou a notícia com o título abaixo:

"Argentina ameaça cassar concessão da Vale"

E no dia 21/03/2013, a Época negócios publicou:

"Justiça argentina acata medida cautelar contra Vale"

Apesar dos títulos das notícias não nos permitir entender o real problema, é possível perceber que durante o período de 12 a 21 de março de 2013 houve um impasse entre a Vale e o governo argentino, que ameaçava a empresa e adotou medidas contra a Vale. Analisando o resultado das ações da Vale nesse período extraídas da base de cotações do UOL acessada em 22 de novembro de 2014, é possível perceber que no acumulado do período de 12 a 21 de março de 2013, as ações caíram 5,78% (disponível em: <http://cotacoes.economia.uol.com.br/acao/cotacoes-historicas.html?codigo=vale5.SA&beginDay=12&beginMonth=3&beginYear=2013&endDay=31&endMonth=3&endYear=2013>. Acesso em: 22 novembro 2014).

A outra palavra que aparece na lista e que gerou curiosidade é "bsgr". Bsgr refere-se à empresa BSG resources da Guiné, parceira da Vale. Ela está associada ao sentimento negativo por conta de alguns problemas que a empresa e a Vale tiveram com o governo da Guiné que acabou pela cassação das concessões das duas empresas em março de 2014. Em 9 de março de 2014, o jornal Estadão publicou:

"Relatório da Guiné pede cassação de concessões da BSGR, parceira da Vale"

Em 17 de março de 2014, o site de notícias Bol publicou a notícia abaixo:

"Guiné aprova proposta de cassar concessões de Vale e BSGR, diz fonte"

Analisando a variação dos preços das ações nesse período extraídas da base de cotações do UOL acessada em 22 de novembro de 2014 é possível perceber que no acumulado do período de 9 a 17 de março de 2014, o valor as ações da Vale caíram 4,03% (disponível em: <http://cotacoes.economia.uol.com.br/acao/cotacoes-historicas.html?codigo=vale5.SA&beginDay=9&beginMonth=3&beginYear=2014&endDay=17&endMonth=3&endYear=2014>. Acesso em: 22 novembro 2014).

Com esses dois exemplos, é possível perceber que as notícias podem ser utilizadas como um sinalizados de curto prazo do valor das ações, confirmando o que já haviam mostrado Mankiw (2010) e Mello e Spolador (2010), Mittermayer (2004), Hagenau *et al*. (2013), que mostraram que a expectativa do mercado pode influenciar o valor das ações. Nos dois casos apresentados, o mercado tinha expectativas negativas sobre a Vale, o que pode justificar a queda do valor das ações.

Portanto, quando apresentadas as listas com as palavras que melhor classificam as notícias em positivas ou negativas, é possível afirmar com uma boa margem de acerto, que essas palavras podem indicar tendência de variação de preço das ações.

5.2.5 CONSIDERAÇÕES FINAIS

Este capítulo apresentou o desenvolvimento do projeto e sua aplicação na classificação das notícias. Foram definidos os parâmetros de pré-processamento que geraram melhores resultados para melhoria da performance do classificador. A Figura 30 mostra o resultado dos primeiros testes que buscaram identificar a melhor parametrização para tokenização.

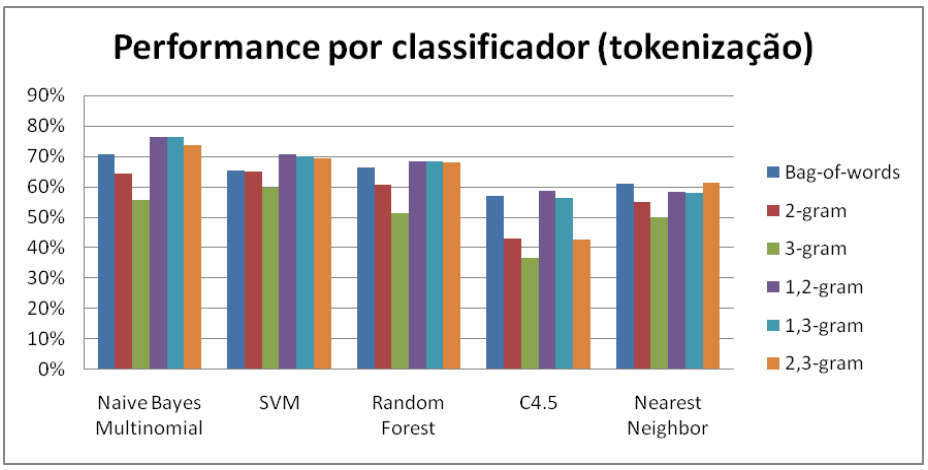

Figura 30 - Performance por classificador (tokenização)
Fonte: Elaborado pelo autor

Para cada classificador de machine learning, foram definidos seis parâmetros de tokenização. Conforme mostra a Figura 30, o que apresentou melhor resultado para a base em estudo foi a utilização do n-gram com n variando entre 1 e 2. Para quatro dos cinco classificadores ele obteve melhor perfomance que os demais. Por outro lado, o n-gram com n igual a 3 obteve pior resultado para todos os classificadores. Isso se deve ao elevado número de atributos irrelevantes gerados por esse tipo de tokenização.

Foi então, adotado a segmentação com *n-gram* 1 e 2 para a realização do segundo grupo de testes, que consistiu em variar a freqüência mínima dos termos presentes na base. A parametrização foi realizada com a freqüência variando entre 1 e 6 vezes. A Figura 31 sintetiza os resultados obtidos. Foram omitidos os dados para freqüência igual a 2 e 4 por conta de seus resultados serem muito semelhantes à freqüência 1 e 3, respectivamente.

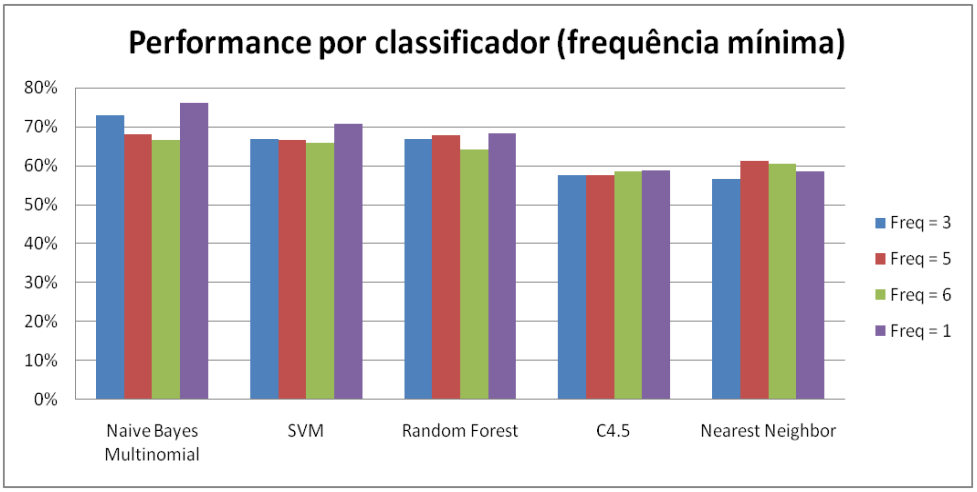

Figura 31 - Performance por classificador (frequência mínima)
Fonte: Elaborado pelo autor

Assim como nos primeiros testes, para cada classificador, a frequência mínima dos termos foi variada seis vezes. A Figura 31 permite concluir que para a base em estudo, a freqüência mínima igual a 1 foi a que apresentou melhores resultados para os classificadores, com exceção do classificador *nearest neighbor*. O aumento do filtro dos atributos pela freqüência mínima dos termos faz com que alguns grupos de palavras relevantes para a classificação das notícias sejam eliminadas, o que justifica a perda de performance.

O último teste dos parâmetros de pré-processamento foi utilizando algoritmos de seleção de atributos. Foram utilizadas duas ferramentas, *correlation feature selection* e qui-quadrado. A Figura 32 mostra a performance obtida para os cinco classificadores gerados utilizando essas duas sistemáticas de seleção de atributos.

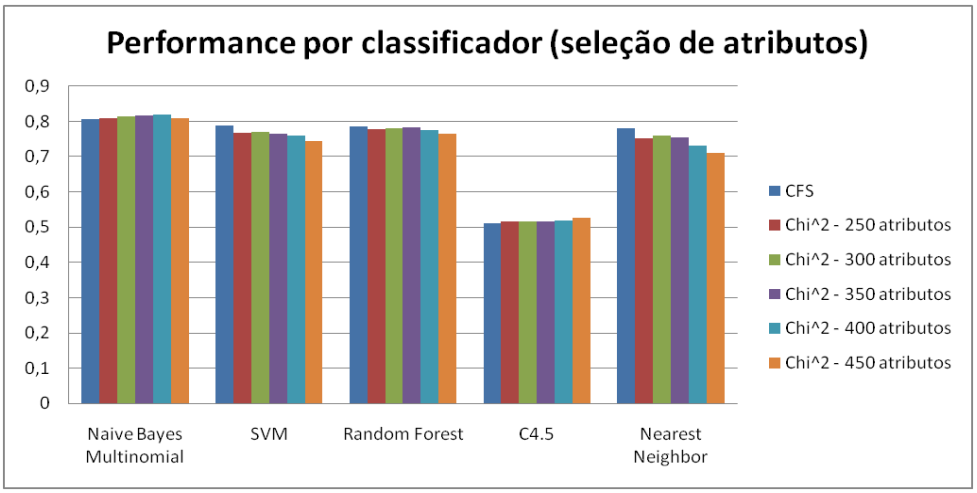

Figura 32 - Perfomance por classificador (seleção de atributos)
Fonte: Elaborado pelo autor

A seleção de atributos gerou melhoria de performance para todos os classificadores. Porém, dependendo do classificador utilizando, uma ferramenta performou melhor do que outra. Para o *Naive Bayes Multinomial* e para o C4.5, o qui-quadrado obteve melhores resultados que o CFS. Já para o SVM, *Random Forest* e o *Nearest Neighbor*, o qui-quadrado apresentou melhor performance. Dessa forma, a escolha da ferramenta de seleção de atributos irá depender do classificador utilizado.

6 CONCLUSÕES

A partir da variação dos parâmetros de pré-processamento, foi possível identificar aqueles que geravam melhor resultado para os classificadores associados ao contexto da Vale e da base de notícias selecionadas. Para processo de tokenização, a abordagem utilizando *n-gram* com n fixo igual a 2 ou 3 se mostrou menos eficiente que *bag-of-words*, que por sua vez se mostrou menos eficiente que *n-gram* com n variável. Apesar de mais complexo e robusto, o *n-gram* com n fixo performou pior que o *bag-of-words*, por conta da grande quantidade de atributos gerados, aumentam-se as possibilidade de combinações, o que dificultou o aprendizado do classificador e com isso, o aumento da taxa de erros e a eficiência, como também apresentou Dumais *et al*. (1998). O melhor parâmetro para *tokenization* foi utilizando *n-gram* com n variando entre 1 e 2.

Quando analisado a frequência mínima de termos, foi possível concluir que o aumento da frequência mínima filtrando somente os atributos mais presentes, não gerou resultados melhores. Isso é explicado pelo fato de que os atributos mais freqüentes não são os que melhor representam a polaridade da notícia. Muitas vezes esses atributos estão associados às duas classes, como é o exemplo da palavra "Vale", que está presente em praticamente todas as notícias e não consegue representar polaridade para as notícias. Ou seja, o filtro dos atributos utilizando a frequência mínima acaba por restringir inúmeros atributos relevantes para a classificação das notícias, por isso a perda de performance.

No que diz respeito à comparação de dois algoritmos de seleção de atributos (qui-quadrado e *correlation feature selection* (CFS)), ambos geraram ganho de performance considerável, o que mostra que na relação de atributos iniciais, haviam muitos redundantes e irrelevantes. Enquanto o CFS se mostrou melhor para três dos cinco classificadores, o qui-quadrado foi melhor para somente um. O outro classificador não obteve resultados satisfatórios com nenhum dos dois métodos. Portanto, é possível afirmar que ambos são relevantes e apesar do CSF ser um modelo mais robusto, a escolha do melhor algoritmo de seleção de atributos depende do classificador utilizado. Para *Support Vector Machine*, *Random Forest* e *K-Nearest Neighbors*, o CSF performou melhor para o estudo analisado, enquanto o qui-quadrado trouxe melhores resultados para o *Naive Bayes Multinomial*.

Comparando-se o desempenho dos cinco algoritmos de aprendizagem utilizados, o que obteve melhor rendimento foi o *Naive Bayes Multinomial*, que dos 20 testes realizados, obteve rendimento superior aos seus concorrentes em 18 vezes. Esse fato chama atenção, pois para os estudos relacionados, apesar de bastante utilizado, geralmente os algoritmos de *Naive*

Bayes eram os que obtinham piores performances quando comparados com outros classificadores. Joachims (1998), Dumais *et al*. (1998), Smailovic *et a*.. (2014), Yang e Liu (1999) mostraram empiricamente que SVM obteve melhores resultados que o *Naïve Bayes*. Porém, o que os demais estudos tem em comum e que diferem desse estudo é que todos foram realizados para a língua inglesa. O estudo, por sua vez, foi realizado em língua portuguesa. Portanto, os resultados obtidos pelo algoritmo *Naive Bayes Multinomial* podem ser um indício que esse modelo é mais indicado para língua portuguesa que os demais. A melhor performance entre todos os classificadores foi obtida utilizando n-*gram* variando entre 1 e 2, remoção de *stopwords*, frequência mínima dos termos igual a um e seleção de atributos qui-quadrado com 400 termos, o qual obteve acurácia superior a 85%, Kappa igual a 0,696 e área ROC igual a 0,924 o que mostra uma concordância substancial entre os valores previstos e os observados, e por ser próxima de um, significa que o classificador está bem representado.

O classificador C4.5 foi o que obteve piores resultados. Seu rendimento foi em geral, mais de 20% inferior aos demais. O baixo valor para o indicador Kappa em praticamente todos os testes revela que esse classificador possui muita aleatoriedade o que não o qualifica como um bom classificador para a situação analisada. Já os outros três obtiveram boa performance, porém inferior ao *Naive Bayes Multinomial*. Um fato que chama atenção é para o classificador *Random Forest*. Da mesma família de classificadores baseados em árvore de decisões como o C4.5, ele obteve resultados muito mais satisfatórios que o C4.5. Isso é justificado principalmente pela utilização dos métodos *bagging* e *boosting*, que reduzem a variação e a aleatoriedade do classificador, gerando melhores resultados.

Foi identificado também a lista de palavras que melhor representam os sentimentos negativos e positivos associados às notícias da Vale e baseado nos estudos de Mankiw (2010) e Mello e Spolador (2010), Mittermayer (2004), Hagenau *et al*. (2013) é possível indicar que essas palavras podem representar uma possível tendência de alta ou baixa das ações da Vale S.A.

Como possíveis trabalhos futuros sugere-se a aplicação dos classificadores gerados para classificação de novas notícias e visando confrontar com o valor da ação e assim mensurar o real impacto das notícias na tendência das ações. É possível repetir o estudo utilizando outros classificadores, de modo a identificar se algum outro classificador obtém melhores resultados que os analisados. É possível realizar estudo também utilizando aprendizagem não supervisionada e assim confrontar com o esse estudo buscando verificar sua eficiência frente à aprendizagem supervisionada. Outra sugestão de trabalho futuro é a utilização de outros parâmetros buscando melhorar a performance dos classificadores ou a

repetição dos mesmos testes realizados utilizando outra base a fim de confirmar se as conclusões geradas nesse trabalho se confirmam para outras bases de dados.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ABDULLAH, M. H. L. B., GANAPATHY, V. Neural network ensemble for financial trend prediction. **Proceedings TENCON**, v. 3, 157-161. 2000

ASSAF NETO, A. **Mercado Financeiro**. 10 ed. São Paulo. Atlas. 2009

BIKHCHANDANI, S.; SHARMA, S. **Herd behavior in financial markets: A review.** International Monetary Funding. 2000

BLANCHARD, O. **Macroeconomia**. 5 ed. São Paulo. Pearson Prentice Hall. 2011.

BUTLER, M.; KESELJ,V. **Financial forecasting using character N-Gram analysis and readability scores of annual reports: Advances in Artificial Inteligence**. Kelowna, Canada. Springer.2009.

CALDEIRA, I. **Banco Central faz novo corte na taxa de juros e Selic vai 7,5% ao ano**. São Paulo: IG, 2012. Disponível em < http://economia.ig.com.br/2012-08-29/banco-central-faz-novo-corte-na-taxa-de-juros-e-selic-vai-a-75-ao-ano.html>. Acesso em 30 jun. 2014.

CAVALCANTE, F., MISUMI, J., RUDGE, L. **Mercado de Capitais: o que é, como funciona**. 7 ed. Rio de Janeiro. Elsevier. 2009.

CHANG, E. C.; CHENG, J. W.; KHORANA, A. An examination of herd behaviour in equity markets: An international perspective**. Journal of Banking and Finance.** v. 24, 1651–1679. 2000.

CHIZZOTTI, A. Pesquisa qualitativa em ciências humanas e sociais. Petrópolis. Vozes. 2006.

CHOWDHURY, G. G. Natural language processing. **Annual Review of Information Science and Technology**. v. 37(1), 51–89. 2003

CHRISTIE, W. G.; HUANG, R. D. Following the pied piper: Do individual returns herd around the market? **Financial Analysts Journal**, v. 51, 31–37. 1995

DAS, S. R.; CHEN, M. Y. Yahoo! For amazon: Sentiment extraction from small talk on the web. **Management Science**. v. 53(9), 1375-1388. 2007.

FELDMAN, S. (1999). Natural language processing in information retrieval : Disponível em: <http://www.scism.lsbu.ac.uk/inmandw/ir/jaberwocky.htm>. Acesso em: 10 jun. 2014, 18:30:00

FELDMAN, R.; SANGER, J. **The Text Mining Handbook – Advanced Approaches in Analyzing Unstructured Data**. New York, USA. Cambridge University Press. 2007

FORMAN, G. An extensive empirical study of feature selection metrics for text classification. **Journal of Machine Learning Research.** v. 3, 1289–1305. 2003

FUNG, G.P.C.; YU, J.X.; YU, X.; LAM, W.. **News Sensitive Stock Trend Prediction**. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Taipei, Taiwan. 2002.

FORTUNA, E. **Mercado Financeiro: produtos e serviços.** 18 ed. Rio de Janeiro. QualityMark. 2011

GIDOFALVI, G. **Using News Articles to Predict Stock Price Movements**. University of California, San Diego: Department of Computer Science and Engineering. 2001

GOMES, H. J. C. **Text mining: análise de sentimentos na classificação de notícias**. Instituto Superior de Estatística e Gestão de Informação. Universidade Nova de Lisboa. Portugal. 2012

GOMIDE, P., MILIDIÚ, R. L. **Assessing Stock Market Time Series Predictors Quality Throug a Paris Trading System**. Eleventh Brasilian Symposium on Neural Networks. São Bernardo do Campo. 2010

GROTH, S.;MUNTERMANN, J. An intraday market risk management approach based on textual analysis. **Decision Support Systems.** v. 50, 680–691, 2011

HAGENAU, M., LIEBMANN, M., NEUMANN, D.. Automated news reading: Stock price prediction based on financial news using context-capturing features. **Decision Support Systems.** v. 55, 685-697. 2013

HUANG, W., GOTO, S., NAKAMURA, M. Decision-making for stock trading based on trading probability by considering whole market movement. **European Journal ofOperational Research**., v. 157, 227-241. 2004

KIM, S.-H., KIM, D., Investor sentiment from internet message postings and the predictability of stock returns. **Journal of Economic Behavior & Organization**. 2014

KUO, R.J; CHEN, C.H.; HWANG, Y.C. An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and articial neural network. **Fuzzy Set and Systems.** , v. 118, 21-45. 2001

LANGLEY, P.; SAGE, S. **Induction of selective Bayesian classifiers.** Proceedings of the Tenth Conference on Uncertain in Artificial Inteligence, Seattle, USA, 1994.

LANGLEY, P.; IBA, W.; THOMPSON, K., **An analysis of Bayesian classifiers**. Proceedings of the Tenth National Conference on Artificial Intelligente. Seatle, USA. 1992

LI, F. The information content of forward-looking statements in corporate filings — a naïve Bayesian machine learning approach. **Journal of Accounting Research**. v. 48 (5), 49–102. 2009

LIU, B. **Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data**. 2. Ed. Springer. New York. EUA. 2009

LINDENBERG, E.; ROSS, S. Tobin's Q Ratio and Industrial Organization. **Journal of Business**. v. 54. 1981

MANKIW, G. **Macroeconomia**. 7. ed. Rio de Janeiro. LTC. 2010.

MARSLAND, S. **Machine Learning: An algorithmic perspective**. 1. ed Champman & Hall. Boca Raton, Florida, EUA. 2009

MARTINEZ, L. C.; da HORA, D. N.; PALLOTI, J. R. M; PAPPA, G. L.; MEIRA JR., W. **From an artificial neural network to a stock market day-trading system: A case study on the BM&F Bovespa.** International Joint Conference on Neural Networks. Atlanta, USA. 2009.

MELLO, P. C. DE., SPOLADOR, H. **Crises financeiras: quebras, medos e especulações de mercado**. 3. ed. São Paulo. Saint Paul. 2010

MITTERMAYER, M.-A**. Forecasting Intraday Stock Price Trends with Text Mining Techniques**. Proceedings of the 37th Hawaii International Conference on Social Systems. Hawaii, USA. 2004

OGURI, P.; MILIDIÍ, R. L.; RENTERIA, R. **Machine Learning for Sentiment Classification.**. MsC Thesis — Department of Informática, Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro. 2006

OH, K. J., KIM, K. J. Analyzing stock market tick data using piecewise nonlinear model. **Expert System with Applications**, v. 22, 249-255. 2002

PINHEIRO, J. **Mercado de Capitais – Fundamentos e Técnicas**. 5 ed. São Paulo. Atlas. 2009

SCHUMAKER, R.P.; CHEN, H. Textual analysis of stock market prediction using breaking financial news: the AZFin text system, **ACMTransactions on Information Systems.** v. 27, 2. 2009.

SMAILOVIC, J.; GRCAR, M.; LAVRAC, N.; ZNIDARSIC, M. Stream-based active learning for sentiment analysis in the financial domain. **Information Sciences**. v.285, 181-203. 2014

TAN,T. Z.; QUEK, C.; NG, G. S. **Brain inspired genetic complementary learning for stock market prediction**. IEEE congress on evolutionary computation, vol. 3, pp. 2653-2660, 2005.

VENTURA, M. O Estudo de Caso como Modalidade de Pesquisa. v.20, 383-386. 2007

WITEN, I.; FRANK, E.; HALL, M. **Data Mining: Practical Machine Learning Tools and Techniques**. 3. Ed. Burlington, USA. Elsevier. 2011

WÜTHRICH, B.; CHO, V; LEUNG, S.; PERMUNETILLEKE, D; SANKARAN, K; ZHANG, J. **Daily stock market forecast from textual web data.** Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, San Diego, USA. 1998

YAO,J.; MA, C.; HE, W. P. Investor herding behaviour of Chinese stock market. **International Review of Economics and Finance.** v. 29, 12–29. 2006

YIN, R. Estudo de caso: planejamento e métodos 2.ed. Porto Alegre. Bookman. 2001

ANEXO A – LISTA DE *STOPWORDS*

De	lhe	hajam
A	deles	houvesse
O	essas	houvéssemos
Que	esses	houvessem
E	pelas	houver
Do	este	houvermos
Da	dele	houverem
em	tu	houverei
um	te	houverá
para	vocês	Houveremos
com	vos	houverão
não	lhes	houveria
uma	meus	houveríamos
os	minhas	houveriam
no	teu	sou
se	tua	somos
na	teus	são
por	tuas	era
mais	nosso	éramos
as	nossa	eram
dos	nossos	fui
como	nossas	foi
mas	dela	fomos
ao	delas	foram
ele	esta	fora
das	estes	fôramos
à	estas	seja
seu	aquele	sejamos
sua	aquela	sejam
ou	aqueles	fosse
quando	aquelas	fôssemos
muito	isto	fossem
nos	aquilo	for
já	estou	formos
eu	está	forem
também	estamos	serei
só	estão	será
pelo	estive	seremos
pela	esteve	serão

até	estivemos	seria
isso	estiveram	seríamos
ela	estava	seriam
entre	estávamos	tenho
depois	estavam	tem
sem	estivera	temos
mesmo	estivéramos	tém
aos	esteja	tinha
seus	estejamos	tínhamos
quem	estejam	tinham
nas	estivesse	tive
me	estivéssemos	teve
esse	estivessem	tivemos
eles	estiver	tiveram
você	estivermos	tivera
essa	estiverem	tivéramos
num	hei	tenha
nem	há	tenhamos
suas	havemos	tenham
meu	hão	tivesse
às	houve	tivéssemos
minha	houvemos	tivessem
numa	houveram	tiver
pelos	houvera	tivermos
elas	houvéramos	tiverem
qual	haja	terei
nós	hajamos	terá
terão	teríamos	teremos
teria	teriam	