PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

DISSERTAÇÃO DE MESTRADO

ALEX SANCHES RODRIGUES

AVALIAÇÃO DE ALGORITMOS PARA ANÁLISE DE SENTIMENTO PARA O PORTUGUÊS BRASILEIRO ESCRITO

Rio de Janeiro
2013

Instituto de Matemática

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI DE APLICAÇÕES E PESQUISAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ALEX SANCHES RODRIGUES

AVALIAÇÃO DE ALGORITMOS PARA ANÁLISE DE SENTIMENTO PARA O PORTUGUÊS BRASILEIRO ESCRITO

Dissertação de Mestrado submetida ao Corpo Docente do Departamento de Ciência da Computação do Instituto de Matemática, e Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Orientador: Carla Amor Divino Moreira Delgado
Co-orientador: Fernando Gil Vianna Resende Junior

Rio de Janeiro
2013

R696 Rodrigues, Alex Sanches

Avaliação de Algoritmos para Análise de Sentimento para o Português Brasileiro Escrito / Alex Sanches Rodrigues. – 2013.
75 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em Informática, Rio de Janeiro, 2013.

Orientador: Carla Amor Divino Moreira Delgado.
Co-orientador: Fernando Gil Vianna Resende Junior.

1. Análise de Sentimentos. 2. Processamento de Linguagem Natural. 3. Naïve Bayes. 4. Máxima Entropia. 5. Support Vector Machine. – Teses. I. Delgado, Carla Amor Divino Moreira (Orient.). II. Resende Junior, Fernando Gil Vianna (Co-orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em Informática. IV. Título

ALEX SANCHES RODRIGUES

Avaliação de Algoritmos para Análise de Sentimento para o Português Brasileiro Escrito

Dissertação de Mestrado submetida ao Corpo Docente do Departamento de Ciência da Computação do Instituto de Matemática, e Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Aprovado em: Rio de Janeiro, 25 de Novembro de 2013.

Prof.ª Carla Amor Divino Moreira Delgado, D.Sc., PPGI/UFRJ (Orientador)

Prof. Fernando Gil Vianna Resende Junior, Ph.D, COPPE/UFRJ (Co-orientador)

Prof.ª Adriana Santarosa Vivacqua, D.Sc., PPGI/UFRJ

Prof.ª Mariane Rembold Petraglia, Ph.D., COPPE/UFRJ

Prof.ª Valeria Menezes Bastos, D.Sc., DCC/UFRJ

Rio de Janeiro
2013

A Deus pelas bençãos diversas que me permitiram chegar até aqui.

AGRADECIMENTOS

Agradeço a meus pais por todos sacrifícios e esforços para que eu pudesse tornar-me um homem melhor.

Agradeço a meus amigos que compreenderam minha ausência e torceram pelo meu sucesso.

Agradeço aos meus orientadores Carla e Fernando, pelos ensinamentos, pela paciência, compreensão e todo o apoio e incentivo durante os trabalhos.

Agradeço em especial à minha noiva Marianna pelo apoio, suporte e incentivo que fez o caminho até aqui ser muito mais prazeroso e gratificante.

RESUMO

Rodrigues, Alex Sanches. **Avaliação de algoritmos para análise de sentimento para o português brasileiro escrito**. 2013. 65 f. Dissertação (Mestrado em Informática) - PPGI, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.

Neste trabalho é apresentado um estudo comparativo de classificadores de sentimento para textos escritos em português brasileiro. O sistema implementado classifica cada frase em uma emoção dentre três: alegria, tristeza, raiva e mais um estado neutro. Foram realizados testes com um corpus anotado com a emoção de cada frase, o qual foi obtido através da transcrição das falas de um filme de ação brasileiro e da anotação por dois voluntários, que tiveram suas anotações cruzadas. Combinações do uso de uni, bi e trigramas em conjunto com ocorrências, contagem de frequências e tf-idf foram usados como extração de características. Durante os testes também foi experimentado o uso de lematização das palavras do corpus. Foram extraídas das falas do filme 2.151 frases e, após o cruzamento das anotações, um total de 1.319 restaram. Os melhores resultados foram de 69,9 , 69,4, 65,7 e 69,3% de precisão, *recall*, *F-score* e acurácia, respectivamente, utilizando diferentes classificadores. Além do conhecimento adquirido sobre desempenho dos algoritmos, os melhores resultados obtidos equivalem-se ou superam, em desempenho, trabalhos semelhantes em português brasileiro.

Palavras-chave: Análise de Sentimentos, Processamento de Linguagem Natural, Naïve Bayes, Máxima Entropia, Support Vector Machine.

ABSTRACT

Rodrigues, Alex Sanches. **Avaliação de algoritmos para análise de sentimento para o português brasileiro escrito**. 2013. 65 f. Dissertação (Mestrado em Informática) - PPGI, Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.

This work presents a comparative study of Brazilian Portuguese written text sentiment classifiers. The implemented system classifies each phrase with respect to three emotions: happiness, sadness, angriness, besides a neutral state. Towards the realization of the tests, a corpus, annotated with each phrase emotion, is proposed. This corpus was obtained by the transcription of a Brazilian action movie lines and an annotation process made by two volunteers. The two anotated corpus went through an agreement check. Combinations of uni, bi and trigrams together with occurrences, frequency and tf-idf were used as feature extraction. One variation tested was the use of lemmatization. From the lines of the movie of choice, it was possible to extract 2,151 phrases and, after the agreement check, a total of 1,319 were left. The best results were 69.9 of precision, 69.4 of recall, 65.7 of F-score and 69.3% of accuracy obtained using different classifiers. Besides the acquired knowledge about the algorithms performance, the obtained results have the same or superior performance if compared to resembling works in Brazilian Portuguese.

Keywords: Sentiment Analysis, Natural Language Processing, Naïve Bayes, Maximum Entropy, Support Vector Machine.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BOW	*bag-of-words*
IIS	*Improved Iterative Scaling*
LSA	*Latent Semantic Analysis*
ME	*Maximum Entropy*
NB	*Naïve Bayes*
NLTK	*Natural Language Toolkit*
RBF	*Radial Basis Function*
SVD	*Singular Value Decomposition*
SVM	*Support Vector Machine*
tf-idf	*term frequency–inverse document frequency*

SUMÁRIO

1 INTRODUÇÃO

A 'análise de sentimentos' (PANG; LEE, 2008), de uma maneira geral, é uma essencial ferramenta para o desenvolvimento da área de interação entre o homem e a máquina. Os trabalhos nessa área são de naturezas diversas, mas um aspecto em particular trata da naturalidade e da humanização do artefato artificial que interage com o ser humano e, nesse tocante, a análise de sentimentos é capaz de apresentar uma significativa contribuição.

Ao tornar um sistema capaz de identificar e utilizar adequadamente as emoções do discurso com a pessoa com quem se está interagindo, cria-se um novo nível de comunicação, permitindo que o sistema seja capaz de captar mais informações e as utilizar de forma a dar uma melhor resposta. A emoção e sutilezas sentimentais fazem parte de uma forma de interação inerente aos seres humanos, mas ainda difícil e com muito a evoluir quanto à utilização por parte dos agentes artificiais que carecem de naturalidade na forma de interagir com os seres humanos.

Este trabalho situa-se nesse contexto por tratar da identificação de emoções em textos, área que tem crescido em número de pesquisas, mas que ainda possui escassez de trabalhos para o idioma português brasileiro. A temática principal dos trabalhos estende-se desde a criação de corpora e léxicos ao desenvolvimento e experimentação de técnicas e algoritmos para a análise de sentimento.

Até o momento, não existem algoritmos nem técnicas consolidadas exclusivamente para a tarefa de identificação de emoções em textos. De uma forma geral, para esse fim são utilizadas técnicas baseadas em regras, algoritmos de aprendizagem de máquina, soluções híbridas e adaptadas, entre outras. Dentre os algoritmos de

aprendizagem de máquina, os mais utilizados são o classificador bayesiano ingênuo (STRAPPARAVA; MIHALCEA, 2008), o de máxima entropia (TRILLA; ALIAS, 2013) e a máquina de vetores de suporte (FERREIRA; DOSCIATTI; PARAISO, 2013), respectivamente em inglês, *Naïve Bayes* (NB), *Maximum Entropy* (ME) e *Support Vector Machine* (SVM).

1.1 Motivação

Dada a escassez de trabalhos de identificação de emoções em textos no idioma português brasileiro, um estudo comparativo voltado ao idioma por si só já constitui um desafio e oferece a possibilidade de importantes contribuições à área no país.

O foco desta pesquisa é a avaliação comparativa das principais técnicas usadas para análise de sentimento, aplicadas ao idioma português brasileiro, e a construção e análise de corpora que viabilize o estudo, promovendo um recurso que permita testes de desempenho dos classificadores e que sirva como base para treinamento daqueles baseados em aprendizagem de máquina.

Uma motivação específica para o desenvolvimento desse trabalho é a possibilidade da aplicação dos resultados na área de síntese de voz, utilizando a saída do sistema de identificação de emoções em texto como parâmetro de entrada de um sistema de síntese de voz com emoções (CRUZ SILVA, 2011).

1.2 Objetivos

Este trabalho trata da identificação, no nível de frases em textos escritos em português brasileiro, de três emoções – alegria, tristeza e raiva – dentre as seis

emoções humanas propostas como universais em (EKMAN, 1993) – alegria, raiva, tristeza, desgosto, medo e surpresa. A ausência de emoção – estado neutro – também é identificada. Nessa dissertação, ao mencionar a anotação e a classificação de emoções, o estado neutro está implicitamente incluído.

O principal objetivo deste trabalho é comparar o desempenho de técnicas e abordagens ao problema do reconhecimento de emoções, aplicados ao português brasileiro. Para alcançar tal objetivo, um sistema deve ser implementado e o mesmo deve ser capaz de utilizar diferentes técnicas e algoritmos, além de permitir o adequado levantamento de desempenho dos classificadores treinados. Os classificadores utilizados foram implementados no *toolkit Scikit-learn* (PEDREGOSA et al., 2011) e parte da extração de características e do processamento do texto foi feito com ferramentas do NLTK (BIRD; KLEIN; LOPER, 2009).

Secundariamente, este trabalho visa à criação de um recurso essencial a esse tipo de estudo, que é um corpus que permita o treinamento e a avaliação dos classificadores. Este corpus deve estar anotado com as emoções que o sistema propõe-se a reconhecer.

Resumidamente, os objetivos desta pesquisa são:

- Criar um corpus em português brasileiro;
- Anotar o corpus frase a frase utilizando as emoções que o sistema propõe-se a reconhecer;
- Implementar um sistema básico que permita o uso de diferentes algoritmos implementados no *Scikit-learn*;
- Realizar testes de desempenho com diferentes algoritmos e abordagens;
- Avaliar os resultados e a influência do corpus.

1.3 Organização da Dissertação

No Capítulo 2 serão apresentados os fundamentos teóricos mais importantes para este trabalho. O Capítulo 3 descreve o sistema implementado e a criação do corpus utilizado. No Capítulo 4 são descritos os testes e apresentados os principais resultados obtidos. As conclusões deste trabalho são expostas no Capítulo 5. Ao fim desta dissertação encontra-se um Apêndice com detalhes e descrição da ferramenta usada como apoio.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são considerados os principais fundamentos teóricos que nortearam esta pesquisa, contendo na Seção 2.1 uma visão geral sobre a área. Na Seção 2.2 é destacada a extração de características e o tratamento dos textos. É feita uma descrição sobre os algoritmos aplicados à análise de sentimentos nesta pesquisa na Seção 2.3. E, por fim, na Seção 2.4 é apresentada uma relação e descrição dos principais trabalhos relacionados.

2.1 Visão Geral

As terminologias de 'análise de sentimentos' ou 'mineração de opiniões' são larga e indistintamente utilizadas para designar o estudo computacional de textos com o objetivo de identificar características interpretativas dos mesmos como as emoções que contêm.

O tipo de estudo mais realizado dentro da análise de sentimentos é o de identificação de opiniões em textos como sendo 'positivas' ou 'negativas', tendo também o estado 'neutro' como variante comumente presente, mas não essencial. Esse tipo de investigação, de identificação de polaridades, tem sido utilizado com finalidades diversas, sendo aplicado para a predição de comportamento de ativos do mercado de ações (SI et al., 2013), avaliação de reação de consumidores a produtos (BAI, 2011) e avaliação da qualidade de atendimento em *call centers* (VAUDABLE; DEVILLERS, 2012), entre outras.

Uma outra vertente da análise de sentimentos, que é a trabalhada nesta pes-

quisa, é o reconhecimento de emoções dentro de um conjunto definido. O conjunto de emoções a ser reconhecido varia de acordo com a aplicação, mas a proposta trabalhada em diversos estudos de Paul Ekman, incluindo (EKMAN, 1993), de que existem seis emoções universais – alegria, raiva, tristeza, desgosto, medo e surpresa – é frequentemente utilizada. Um subconjunto também utilizado em alguns trabalhos, como em (KIM; VALITUTTI, 2010), é o que engloba apenas a alegria, a tristeza e a raiva, que é o subconjunto utilizado neste trabalho por ser reduzido, de ocorrência comum e facilmente identificável. Assim como na classificação de polaridades, o estado neutro também pode fazer parte do conjunto de classificações possíveis.

O tipo dos corpora utilizados é diversificado, variando em tamanho e na origem do texto. Comentários em *blogs*, na rede social *twitter* e chamadas de notícias de jornal são tipos de texto usados frequentemente pela sua facilidade de aquisição e desenvolvimento de um corpus de tamanho adequado de maneira semi-automática. Alguns exemplos de trabalhos que usaram os citados como corpus são (SI et al., 2013), (MARTÍNEZ-CÁMARA et al., 2013) e (BELLEGARDA, 2011).

Uma parte essencial para o bom desempenho de um sistema de análise de sentimentos reside no tratamento do texto e na extração de características do mesmo. Duas evidências recentes da importância dessa parte do processo podem ser encontradas em (FERREIRA; DOSCIATTI; PARAISO, 2013) e (WANG et al., 2011).

O componente que efetivamente realiza a identificação das classes do sistema de análise de sentimentos ou mineração de opiniões é o classificador. É o classificador quem efetua a tarefa de inferir, a partir do texto dado como entrada, qual a polaridade ou a emoção daquela frase, parágrafo ou documento. O desenvolvimento de classificadores ou experimentações com métodos já conhecidos e estabelecidos é atividade recorrente. Diversos tipos de classificadores são utilizados e estudados, desde classificadores baseados em palavras-chave e regras linguísticas criadas manualmente

por um especialista, quanto a métodos de aprendizado de máquina e híbridos. Em (BELLEGARDA, 2011), (AGRAWAL; AN, 2012), (QUAN; REN, 2009) e (PRABOWO; THELWALL, 2009) pode ser observada a utilização de distintos tipos de classificadores.

No Brasil, o cenário de pesquisa na área ainda é incipiente. Em (PARDO et al., 2010) toda a área de linguística computacional e processamento de linguagem natural é considerada ainda emergente por motivos que variam desde a falta de financiamento, passando pela falta de recursos linguísticos para o português brasileiro, até a dificuldade de conseguir colaboração de mais e novos pesquisadores.

Na Figura 2.1 pode ser visualizado o esquema de funcionamento típico de um sistema de classificação de sentimentos que utiliza algoritmos de aprendizado de máquina.

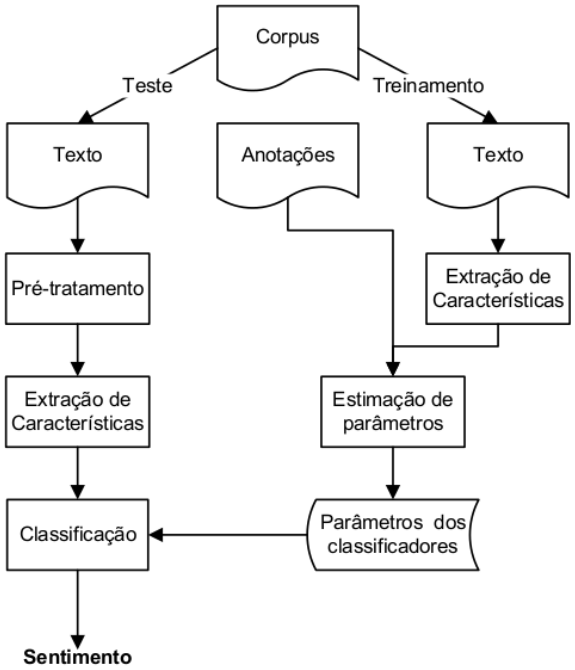

Figura 2.1: Fluxograma típico de um sistema que usa algoritmos de aprendizado de máquina.

2.2 Extração de Características

A extração de características de um texto pode ser compreendida como o processo de transformação do texto puro e simples em um conjunto de características interpretáveis por um classificador. O texto, em forma de lista de palavras, frases, parágrafos, etc., ordenados como originalmente escrito, que possui sentido significativo a quem possa ler, passa a ser representado de forma que, normalmente, apesar de não fazer mais sentido à leitura por uma pessoa, tem agregado um conjunto de informações que possibilitam seu processamento por classificadores computacionais de maneira adequada além de ser mais simples de ser manipulado. Um exemplo seria a possibilidade de representar a frase "Processamento de Linguagem Natural é multidisciplinar." como "de é Linguagem multidisciplinar Natural Processamento", que é o resultado do ordenamento das palavras da frase em ordem alfabética.

Os processos de extração de características utilizados neste trabalho de pesquisa foram diversos e simples. O conceito principal utilizado foi o de *bag-of-words* (BOW), que consiste em desconsiderar a ordem das palavras de um texto e tratá-lo como um simples "saco-de-palavras". Essa abordagem claramente despreza a informação contida na ordem sequencial das palavras de um texto, porém, apesar de sua ideia ser, aparentemente, deficiente, é amplamente utilizada na literatura. Além do BOW, outro importante conceito utilizado foi o de n-gramas. Os n-gramas consistem na definição sistemática de unidades de trabalho com um tamanho n definido. Foram utilizados três tipos de n-gramas diferentes e suas combinações: os unigramas, os bigramas e os trigramas. Os três consistem em dividir o texto em blocos de uma, duas e três palavras consecutivas, respectivamente. A utilização de unigramas é importante porque é a representação unitária mais básica de um texto, já que cada palavra possui seu valor e contribui com seu peso para o sentido do texto. Bigramas, trigramas e outros n-gramas com valores de n maiores, são importantes por possibilitarem que expressões que possuem mais de uma palavra mantenham-se

inteiras. Por exemplo, “Análise de Sentimento” é uma expressão única mas que só mantém-se inteira quando representada por um trigrama. Fazendo uso, novamente, da frase “Processamento de Linguagem Natural é multidisciplinar” como exemplo, os n-gramas utilizados no trabalho, aplicados a essa frase seriam:

- Unigramas: ‘Processamento’, ‘de’, ‘Linguagem’, ‘Natural, ‘é’ e ‘multidisciplinar’;
- Bigramas: ‘Processamento de’, ‘de Linguagem’, ‘Linguagem Natural’, ‘Natural é’ e ‘é Multidisciplinar’;
- Trigramas: ‘Processamento de Linguagem’, ‘de Linguagem Natural’, ‘Linguagem Natural é’ e ‘Natural é Multidisciplinar’.

A segunda parte da extração de características que foi utilizada foi a vetorização e quantificação dos n-gramas. Essa operação utilizou três variantes. A primeira consiste em considerar apenas as ocorrências ou não dos n-gramas, portanto permitindo uma representação binária, indicando a existência ou ausência de determinado n-grama em um texto. A segunda representação foi através da contagem de frequências, que consiste apenas em contar o número de vezes em que um n-grama aparece no texto. A terceira quantificação utilizada foi a tf-idf, do inglês *term frequency - inverse document frenquency*, que procura mensurar a importância de um termo para um texto dentro de um corpus. É uma medida comum de ser utilizada para evitar valorizar um termo muito frequente para todas as classes que se deseja classificar, o que sugere que não seja um termo que adicione tanto valor e/ou significado àquela parte do texto em específico. O cálculo do tf-idf é feito através do produto $TF \times IDF$. Definidos T_j como um texto, t_i como um termo:

$$TF : tf_{i,j} = \frac{\text{frequência de } t_i \text{ em } T_j}{\text{maior frequência de um termo em } T_j} \tag{2.1}$$

$$IDF : idf_i = \log \frac{\#\text{total de textos do corpus}}{\#\text{textos que contêm } t_i} \tag{2.2}$$

Em (SALTON; BUCKLEY, 1988) há uma discussão mais aprofundada sobre a questão.

2.2.1 Lematização

Um tipo de processamento passível de ser realizado com um texto, e que foi experimentado neste trabalho, é a lematização. A lematização é uma prática que tem seu funcionamento intimamente ligado ao idioma do texto em que é aplicado, uma vez que consiste na redução de palavras a seu radical primitivo. Essa prática elimina as variações de uma palavra, facilitando o processamento com vistas a melhorar o resultado, contando com a ideia de que uma palavra mantém seu sentido independente da variante. Um exemplo descrito em (MARTINAZZO, 2010) é o de "terra, terrinha, terriola, térreo, terráqueo, terreno, terreiro, terroso" que possuem o mesmo radical 'terr-'.

O algoritmo que foi aplicado neste trabalho foi a versão implementada no *scikit-learn* do proposto para o português em (ORENGO; HUYCK, 2001), que é um removedor de sufixos baseado em 8 passos, cada passo sendo composto por um conjunto de regras, totalizando 199 regras criadas tendo como referência os sufixos mais presentes na língua portuguesa. Os oito passos podem ser visualizados na Figura 2.2, adaptada de (ORENGO; HUYCK, 2001).

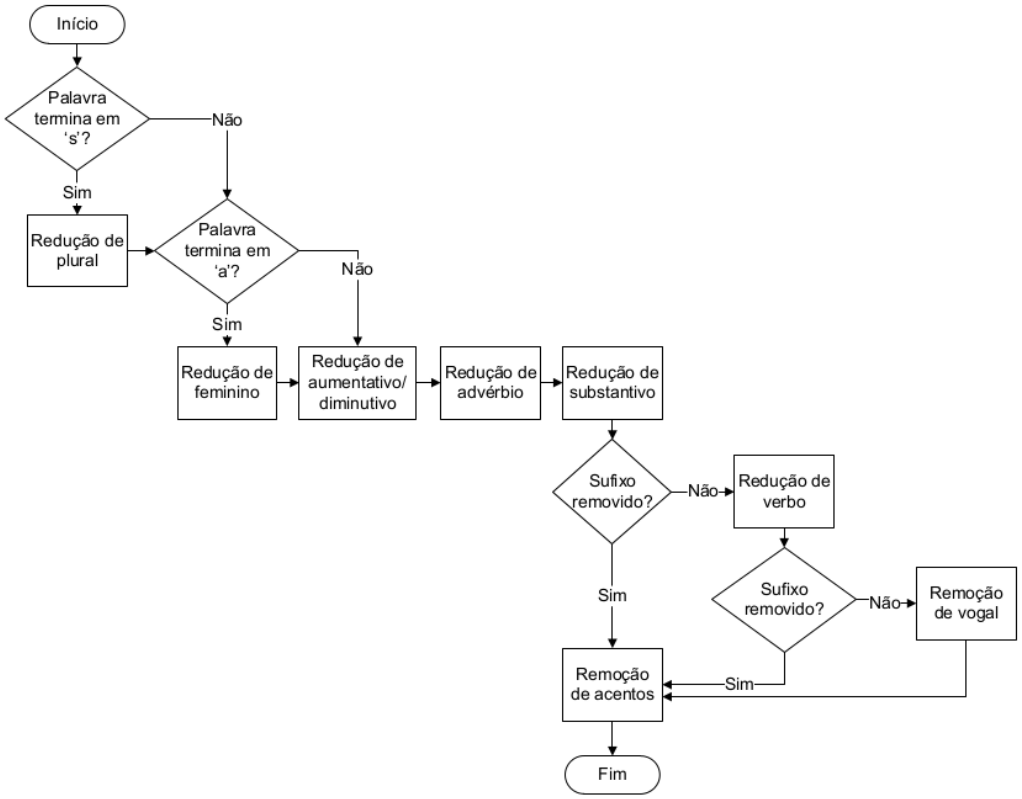

Figura 2.2: Fluxograma traduzido com os 8 passos do lematizador. Figura original em (ORENGO; HUYCK, 2001).

2.3 Algoritmos

Virtualmente, qualquer algoritmo apto a fazer classificação a partir de conjuntos de características é aplicável à análise de sentimentos, porém, dentro da área, três algoritmos de aprendizado de máquina são frequentemente mais usados, que são os classificadores NB, ME e SVM, como usados em (PANG; LEE; VAITHYANATHAN, 2002).

2.3.1 Naïve Bayes

Contextualizando para o universo da análise de sentimentos, um classificador NB calcula a probabilidade $p(C|T)$ de um texto ter uma determinada classificação, considerando um conjunto de $|C|$ classes $C = \left\{c_1, c_2, \ldots, c_{|C|}\right\}$ e outro de textos, com cada texto T sendo $T = \{t_1, t_2, \ldots, t_n\}$, representando sua constituição por um conjunto de n características obtidas a partir da extração, podendo ser, por exemplo, o conjunto de unigramas, bigramas ou trigramas.

Pelo teorema de Bayes, $p(C|T)$ pode ser calculada segundo:

$$p(C = c_j|T) = \frac{p(T|C = c_j)p(C = c_j)}{p(T)}. \tag{2.3}$$

A partir de 2.3, expandindo, obtém-se:

$$p(C = c_j|t_1, t_2, \ldots, t_n) = \frac{p(t_1, t_2, \ldots, t_n|C = c_j)p(C = c_j)}{p(t_1, t_2, \ldots, t_n)} \tag{2.4}$$

Assumindo que as características sejam independentes entre si, então

$$p(t_i|C = c_j, t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) = p(t_i|C = c_j) \tag{2.5}$$

vale para qualquer i, daí, aplicando em 2.4 e simplificando, obtém-se:

$$p(C = c_j|t_1, t_2, \ldots, t_n) = \frac{p(C = c_j) \prod_{i=1}^{n} p(t_i|C = c_j)}{p(t_1, t_2, \ldots, t_n)}. \tag{2.6}$$

Como $p(t_1, t_2, \ldots, t_n)$ é constante para qualquer classe $C = c_j$, calculando

$$c = \arg\max_{c_j} p(C = c_j) \prod_{i=1}^{n} p(t_i|C = c_j) \tag{2.7}$$

encontra-se a classe mais provável.

O treinamento do classificador consiste na estimação de $p(C = c_j) \forall j \in C$ e da distribuição de $p(t_i|C = c_j)$ a partir de um conjunto de dados. O conjunto de dados de treinamento para a tarefa de análise de sentimentos deve ser um corpus anotado segundo as classificações que se deseja realizar e com as características pretendidas já extraídas.

A partir do corpus de treinamento pode-se calcular $p(C = c_j)$ de maneira simples da seguinte forma:

$$p(C = c_j) = \frac{\#\text{ocorrências de } c_j}{\text{tamanho do corpus}}. \tag{2.8}$$

O cálculo das probabilidades condicionais $p(t_i|C = c_j)$ também pode ser realizado de maneira simples da seguinte forma:

$$p(t_i|C = c_j) = \frac{\#\text{ocorrências de } t_i \text{ no corpus quando } C = c_j}{\#\text{total de textos classificados como } C = c_j}. \tag{2.9}$$

Ambas maneiras de estimação das probabilidades que foram apresentadas possuem a vantagem de serem fáceis de computar, porém possuem o problema de, em não havendo uma instância da classe ou do termo dentro da condição da contagem, a probabilidade estimada ser 0, anulando $p(C = c_j) \prod_{i=1}^{n} p(t_i|C = c_j)$. Para

o caso em que $p(C = c_j)$ seja estimada como 0, não há maiores implicações negativas, uma vez que, se uma classe não aparece em nenhuma instância do corpus de treinamento, ela deve ser corretamente desconsiderada, lembrando também que impossibilitaria o cálculo de $p(t_i|C = c_j)$. Quanto a estimar $p(t_i|C = c_j)$ como 0, pode implicar em zerar o produto porque um único termo deixou de aparecer no corpus de treinamento com uma determinada classificação e esse comportamento é indesejado por apenas permitir a classificação de novos textos com termos presentes no corpus de treinamento. Para contornar esse problema, é comum que seja usada uma forma de suavização, normalmente, adicionando uma constante α ao cálculo em 2.8. Os casos em que $\alpha = 1$ são chamados de suavização de Laplace e quando $0 < \alpha \leq 1$ são chamados de suavização de Lidstone.

A escolha do tipo de distribuição para $p(t_i|C = c_j)$, no universo da análise de sentimentos, é, comumente, feita entre a de Bernoulli e a multinomial que partem de que t_i tenha valores discretos. Uma discussão mais aprofundada pode ser encontrada em (METSIS; ANDROUTSOPOULOS; PALIOURAS, 2006).

2.3.2 Máxima Entropia

O classificador ME (NIGAM; LAFFERTY; MCCALLUM, 1999) também realiza a classificação através do cálculo da probabilidade $p(C = c_j|T)$, usando, para isso, a fórmula exponencial:

$$p(C = c_j|T) = \frac{1}{Z(T)} \exp\left(\sum_i \lambda_i F_i(T, C = c_j)\right), \tag{2.10}$$

onde $F_i(T, C = c_j)$ é uma característica e λ_i é um parâmetro de peso a ser estimado. $Z(T)$ é um fator de normalização da forma:

$$Z(T) = \sum_j \exp\left(\sum_i \lambda_i F_i(T, C = c_j)\right). \tag{2.11}$$

Para a estimação dos parâmetros do classificador é utilizado o algoritmo *improved iterative scale* (IIS), que utiliza a estratégia de *hillclimbing*. O algoritmo deve encontrar incrementalmente o conjunto mais provável de parâmetros até convergir. Mais detalhes sobre a estimação dos parâmetros e treinamento do modelo podem ser encontrados em (NIGAM; LAFFERTY; MCCALLUM, 1999) e (BERGER; PIETRA; PIETRA, 1996).

2.3.3 SVM

O funcionamento básico de um classificador SVM consiste em construir um hiperespaço tomando as características extraídas como coordenadas e, após o devido processo de treinamento, utilizar hiperplanos para dividir o hiperespaço de forma a separar as classes que se deseja classificar. O SVM busca a separação ótima entre classes, maximizando uma função de margem, que calcula a distância entre o hiperplano separador e os pontos mais próximos das classes separadas.

Sendo o hiperplano de separação representado pelo vetor $\vec{w}$, o problema pode ser entendido como de otimização com restrições. Com $c_j \in \{1, -1\}$ sendo a correta classificação de um texto T_j, a solução pode ser descrita como:

$$\vec{w} = \sum_j \alpha_j c_j \vec{T_j}, \alpha_j \geq 0, \tag{2.12}$$

onde α_j é obtido através da otimização de um problema dual e $\vec{T_j}$ é a representação vetorial do texto T_j.

Dada sua fundamentação, o SVM é capaz de realizar a separação entre apenas duas classes de cada vez; porém dois tipos de adaptações para separação em mais classes podem ser realizadas. A primeira é a criação de vários classificadores do tipo *um vs resto*, onde para cada classe c_j haveria um classificador para separar c_j das

outras classes. A segunda possibilidade é a criação de classificadores do tipo *um vs um*, onde para cada classe c_j, haveria $|C| - 1$ classificadores. Para a determinação de qual classe é a correta, a função de saída de maior valor entre os classificadores é a utilizada. Em (CRAMMER; SINGER, 2002) é apresentada uma discussão sobre utilização de SVM para problemas com mais de duas classes.

O tipo de separação mais simples é a linear, porém outras também são utilizadas, uma vez que não se pode esperar que os dados sempre sejam linearmente separáveis. Para isto são utilizam-se funções chamadas de funções de kernel. As funções de kernel não-lineares mais comuns são as do tipo polinomial, quadrática, hiperbólica e *radial basis function* (RBF).

Detalhes sobre o algoritmo podem ser encontrados em (CORTES; VAPNIK, 1995).

2.4 Trabalhos Relacionados

Como já mencionado, a área do tema desta dissertação ainda é incipiente no Brasil e poucos são os trabalhos publicados sobre o tema. Em termos específicos de trabalhos de identificação de emoções para frases escritas em português brasileiro apenas dois trabalhos publicados foram encontrados.

Em (MARTINAZZO, 2010), o trabalho consiste em uma pesquisa de mestrado onde a autora desenvolve um sistema para classificação de textos curtos, construindo um corpus a partir de notícias de um sítio da *internet* onde cada entrada do corpus consistiu do título da notícia, juntamente com um pequeno texto descritivo do conteúdo que segue o título. A autora utilizou o classificador do tipo *latent semantic analysis* LSA e aplicou-o ao corpus com 1.000 notícias e obteve um desem-

penho que teve uma acurácia que variou entre 72% e 41,43%, dependo do método de avaliação. O método de avaliação não seguiu o padrão dos trabalhos da área e fez uma avaliação *a posteriori* das classificações do sistema, realizando, na prática, uma análise de concordância com as anotações do algoritmo, método que pode ser considerado tendencioso.

Em (FERREIRA; DOSCIATTI; PARAISO, 2013) o foco do trabalho é a proposta de método para pré-processamento de textos com o objetivo de identificarem-se as emoções dos mesmos em um momento posterior. O classificador utilizado pelos autores foi o SVM e foi aplicado em um corpus com 1.050 textos, com uma divisão de 70% a ser usada para treinamento e o restante para testar o sistema. Apesar de não descrever com detalhes o processo de avaliação, o resultado apresentado foi de acurácia de 48%.

Já quanto aos trabalhos de identificação de emoções em línguas estrangeiras que classificam textos frase a frase, em (STRAPPARAVA; MIHALCEA, 2008), vários métodos são utilizados e comparados. Diferentes algoritmos destacam-se em cada métrica de avaliação. Ao fim, os melhores resultados obtidos foram de 38,28 de precisão, 90,22 de *recall* e 17,57 de *F-score*, valendo ressaltar que o corpus foi formado por 8.761 postagens de *blogs*.

Utilizando a mesma técnica que (MARTINAZZO, 2010) como base, (BELLEGARDA, 2011) aplica seu sistema a um corpus de 1.250 manchetes de notícias e obtém resultados de 20,9 de precisão, 91,7 de *recall* e 34,0 de *F-score*. Vale salientar que o foco do trabalho foi um melhor rendimento em termos de *recall*. O mesmo trabalho teve seus resultados reconfirmados em (BELLEGARDA, 2013).

3 IMPLEMENTAÇÃO

Neste capítulo é dada uma descrição detalhada do que foi desenvolvido para este trabalho. Na Seção 3.1 é descrita a implementação do sistema utilizado para a realização das classificações. Na Seção 3.2 é explicada a construção do corpus utilizado e seu processo de anotação.

3.1 Sistema

O sistema implementado foi dividido em duas grandes partes, uma que trata o texto e faz a extração de características e outra que utiliza as técnicas de aprendizado de máquina para a construção dos classificadores. Dois *toolkits* foram utilizados, o *Natural Language Toolkit* (NLTK) (BIRD; KLEIN; LOPER, 2009) e o *Scikit-learn* (PEDREGOSA et al., 2011), o primeiro para auxílio na manipulação do texto e o segundo para a utilização das técnicas e aplicação no contexto do trabalho.

Para a construção do sistema, foram utilizadas técnicas de aprendizado de máquina comumente empregadas na área juntamente com outras não usualmente aplicadas com esse fim. Ao todo foram aplicadas e testadas nove técnicas de aprendizado e algumas variações do classificador bayesiano ingênuo, máxima entropia, árvore de decisão, SVM e *Linear Discriminant Analysis* (LDA) (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

A parte de extração de características consistiu em utilizar combinações entre uni, bi e trigramas; ocorrências, contagem de frequências e tf-idf foram exploradas. Adicionalmente foi testado o uso de lematização das palavras do texto. O corpus

foi explorado como um conjunto de frases, onde cada frase foi representada por um vetor numérico a ser utilizado pelos classificadores.

3.1.1 Extração de Características

O primeiro tipo de extração de características utilizado foi a contagem de frequências. A contagem de frequências consiste em, para cada unidade textual (no nosso caso cada unidade textual é uma frase) a ser avaliada, cada ocorrência dos n-gramas dessa unidade textual é contada e inserida no vetor numérico. Esse tipo de extração melhor representa grandes unidades textuais como parágrafos, *emails*, reportagens de jornal, capítulos de livro ou documentos inteiros em geral.

A extração de características segundo as ocorrências é semelhante ao da contagem de frequências, diferindo apenas no valor a ser inserido no vetor numérico, que passa a ser apenas 1 ou 0, representando, respectivamente, a ocorrência ou não de um determinado n-grama na unidade textual usada. Para o caso deste estudo, esse tipo de extração tem uma equivalência muito próxima ao da extração de frequências, já que a repetição de n-gramas em uma mesma frase não ocorre tão recorrentemente.

Um terceiro tipo de extração de características utilizado largamente nos trabalhos da área e implantado nesse trabalho é o tf-idf, que representa as unidades textuais através de um cálculo de frequências inversas onde procura-se relativizar as importâncias da ocorrência de um determinado n-grama em sua unidade textual através da comparação entre o número de ocorrências desse n-grama em sua unidade textual e em todo o universo de unidades textuais que fazem parte do mesmo corpus. Essa técnica visa reduzir a importância na consideração de n-gramas mais corriqueiros, como, por exemplo, no caso unigramas, o uso de artigos: a, o, um, uma, etc.. Uma prática comum, que também tem o objetivo de diminuir o impacto

de palavras muito comuns, é o uso de *stop words*, que consiste em uma lista das palavras mais comuns de uma língua, onde essas *stop words* são retiradas do texto. Para a extração desse tipo de característica foi utilizado o *Scikit-learn*.

Em conjunto com as extrações descritas, variações na seleção da n-gramas foram testadas. O caso base testado foi a aplicação em unigramas, ou seja, cada palavra foi considerada para a formação dos vetores de ocorrências, frequências e tf-idf. A utilização de bi e trigramas consistiu em agrupar as palavras de cada frase em pares e trios, respectivamente, e aplicar essa subunidade da mesma forma que os unigramas. A junção de duas ou mais dessas subunidades também foi testada. Para a extração de bi e trigramas foi utilizado o NLTK.

3.1.2 Classificadores

Com o auxílio do *Scikit-learn*, nove diferentes técnicas de aprendizado de máquina foram testadas. A escolha do *toolkit* deu-se pela facilidade de utilização e eficiência computacional da implementação dos classificadores, permitindo maior dedicação à execução de experimentos e possibilitando o teste de desempenho de algumas técnicas não-usualmente aplicadas para os fins deste trabalho.

O classificador bayesiano ingênuo, um dos mais utilizados para classificação de textos, foi aqui aplicado usando duas variações. Ambas as variações diferem quanto à distribuição estatística esperada que melhor representa os dados e aqui foram testados os casos de distribuição multinomial e de Bernoulli.

Outra técnica amplamente utilizada com o propósito de classificação de textos é o SVM, que foi testado em três variações. O SVM foi aplicado utilizando funções de kernel RBF e linear e uma terceira variação, daqui em diante representada apenas

como Nu, oferecida pelo *toolkit* que define um valor que serve como limite superior para a fração de erros de treinamento e como limite inferior para a fração de vetores de suporte.

Os métodos de árvores de decisão e máxima entropia podem ser utilizados tanto para problemas de classificação quanto de regressão e também são utilizados em uma grande quantidade de trabalhos.

Dois métodos utilizados em caráter puramente experimental foram os de *Extremely Randomized Trees* (GEURTS; ERNST; WEHENKEL, 2006) e LDA. A utilização destes métodos deu-se principalmente pela facilidade para aplicação proporcionada pelo uso do *toolkit*.

Não fez parte da etapa de implementação do sistema a definição de quais são os parâmetros que resultam em melhor desempenho por parte de cada um dos algoritmos; todos eles foram utilizados com seus valores padrão definidos no *toolkit*, que por sua vez foram baseados nos seguintes trabalhos (ZHANG, 2004), (MCCALLUM; NIGAM et al., 1998),(METSIS; ANDROUTSOPOULOS; PALIOURAS, 2006), (FAN et al., 2008), (CORTES; VAPNIK, 1995), (GUYON; BOSER; VAPNIK, 1993), (HASTIE; TIBSHIRANI; FRIEDMAN, 2001), (OLSHEN; STONE, 1984), (QUINLAN, 1993) e (GEURTS; ERNST; WEHENKEL, 2006).

3.2 Corpus Proposto

Dada a ausência de um corpus disponível no português do Brasil, anotado frase a frase com as emoções utilizadas neste trabalho, identificou-se a necessidade da criação de um novo corpus, específico para essa aplicação.

Para a criação do corpus foi escolhido fazer a transcrição das falas de um filme de ação brasileiro, que se passa em uma favela carioca e mostra o crescimento e evolução dos atores e da favela. A escolha desta opção foi devido à natureza da anotação a ser utilizada. Em tratando-se de anotações de emoções, é necessário compreender a subjetividade do texto além de realizar-se adequada avaliação do contexto em que está inserido. Além disso, é também importante lembrar, que, como a avaliação do conteúdo emocional é subjetiva, a interpretação das emoções identificadas está altamente dependente da capacidade de avaliação e compreensão do anotador, além de seu estado emocional e humor no momento da anotação. Portanto, visando diminuir a dependência de todos esses fatores, decidiu-se utilizar as falas de um filme, que por si só já são interpretadas pelos próprios atores com uma determinada emoção. A utilização dessa abordagem permite diminuir o grau de dependência por parte do anotador, que agora deve identificar a emoção interpretada por um ator. Essa avaliação possui suas sutilezas e dificuldades também, mas é menos subjetiva que a inferência de uma emoção a partir apenas de um texto.

A escolha de um filme de ação brasileiro e o filme especificamente escolhido deu-se por dois principais fatores. O primeiro fator é a independência da tradução e a manutenção da originalidade do texto, pois, caso fosse utilizado o texto traduzido de filme estrangeiro, esses elementos, que certamente possuem muita informação emocional, seriam perdidos ou até mesmo corrompidos. O segundo fator é a escolha por um filme de ação, que possui tradicionalmente muitas falas com conteúdo dentro do conjunto das emoções escolhidas. A possível seleção de um filme de outro gênero poderia incorrer em um grande desbalanceamento entre frases com uma emoção em relação às outras.

3.2.1 Processo de Anotação

Inicialmente, o filme de escolha teve suas falas claras integralmente transcritas. Casos onde havia sobreposição ou confusão de falas entre os personagens foram tratados através da separação das falas dos atores ou assumindo apenas a fala dominante, sendo esta a mais clara ou com maior conteúdo textual. Contrações usuais à fala, como dizer "tá" ao invés de "está", foram transcritos em sua forma completa. Nenhuma outra alteração foi feita na transcrição a fim de manter, ao máximo, a originalidade do texto. O filme foi anotado por dois anotadores, o primeiro também foi o responsável pela transcrição das falas e sua revisão, enquanto que o segundo apenas realizou anotações.

No processo de anotação os anotadores assistiram ao filme, separadamente um do outro, interpretando a emoção passada pelos atores em cada uma das frases transcritas. Ambos foram instruídos a considerar apenas a percepção emocional transmitida pelo ator em sua interpretação, evitando levar em consideração seus sentimentos e impressões pessoais quanto ao texto e ao filme propriamente dito.

O processo de anotação foi definido rigidamente permitindo apenas a anotação das três emoções de alegria, tristeza e raiva, além de um estado extra que representa a ausência de emoção. Em caso de dúvida, o procedimento adotado foi de que se assistisse repetidamente a passagem em questão até que houvesse segurança quanto à interpretação do trecho e, consequentemente, da anotação.

Ambos anotadores responsabilizaram-se por revisar suas anotações, realizando modificações caso fosse julgado necessário. Para ambos anotadores uma pequena porção das anotações foi alterada mediante novo processo de audiência do filme.

Ao final, 2151 frases foram anotadas por cada um, gerando dois corpus, cada qual com as anotações de seu respectivo anotador. Os números e a distribuição das anotações podem ser observados na Tabela 3.1. Ao realizar o cruzamento das anotações, obteve-se 61,3% de concordância, com 1319 frases anotadas igualmente pelos dois voluntários. A partir das frases onde houve concordância entre ambos, foi composto um terceiro corpus. O resultado do cruzamento é exibido na Tabela 3.2.

Tabela 3.1: Resultados do processo de anotação.

	Anotador 1	Anotador 2
Alegria	419	242
Tristeza	344	64
Raiva	596	557
Neutra	792	1288

Tabela 3.2: Matriz de confusão das anotações sendo o primeiro anotador representado pelas linhas e o segundo pelas colunas.

	Alegria	Tristeza	Raiva	Neutra
Alegria	177	4	9	229
Tristeza	5	38	38	263
Raiva	14	6	442	134
Neutra	46	16	68	662

Observando as Tabelas 3.1 e 3.2 é possível identificar a existência de tendências distintas dos anotadores. Comparando as anotações dos voluntários, o segundo anotador mostra maior tendência em classificar as falas como neutras, explicando os altos números de discordância na coluna ‘Neutra’ da matriz de confusão.

4 TESTES

Neste capítulo são apresentados a metodologia, os resultados e a avaliação dos testes realizados. Na Seção 4.1 é descrita a metodologia utilizada durantes os experimentos. Na Seção 4.2 são exibidos os resultados obtidos durante os testes e na Seção 4.3 a avaliação dos mesmos.

4.1 Metodologia

Como metodologia básica de testes foi adotado o método de amostragem aleatória com 30 repetições para cada caso de teste. Os casos de teste básicos consistiram na aplicação das nove técnicas enunciadas aplicadas à extração de características baseada em unigramas, ocorrências, contagem de frequências e tf-idf com variação da utilização ou não de lematização, totalizando 6 casos básicos de teste. Cada caso básico de teste foi avaliado utilizando conjuntos de treinamento com 50, 60, 70, 80 e 90% das frases.

O processo de amostragem consistia em uma seleção aleatória das frases anotadas com cada emoção, para garantir a proporcionalidade original do texto nos conjuntos de treinamento e de teste. Cada rodada de teste consistiu na amostragem, teste de cada um dos algoritmos e registro dos resultados.

Inicialmente foi feita uma série de testes com o corpus resultante das anotações apenas do anotador 1. Após a primeira bateria de testes, os 5 algoritmos que apresentaram os melhores resultados foram selecionados para serem avaliados em casos de teste com as extrações de características que permitiram o desempe-

nho superior, complementadas com a utilização de bigramas e trigramas. Além da qualidade dos resultados, outro fator para a escolha de 5 algoritmos para testes posteriores é o tempo de execução de cada caso de testes. Em média, o tempo de execução de todos os algoritmos, com 30 repetições, para um tipo de extração de características e um tamanho de conjunto de treinamento foi da ordem de 30 horas. Em um segundo momento, outro corte foi feito e apenas os três melhores algoritmos passaram a ser testados.

Ao fim do processo, o corpus com as anotações resultantes da concordância dos dois anotadores foi utilizado para mais uma bateria de testes.

4.2 Resultados

Para a avaliação dos resultados foram utilizadas as médias obtidas em cada métrica, sem considerar o desvio padrão na comparação.

Após um total de 900 testes com casos de conjunto de treinamento com 90% do total de frases, foi possível observar que os classificadores bayesiano ingênuo, de máxima entropia e SVM tiveram desempenhos sensivelmente superiores aos outros que foram avaliados. Deve-se ressaltar que o algoritmo SVM com função de kernel RBF apresentou desempenho inferior.

Dentre os casos avaliados foi possível destacar três extrações de características que permitiram os melhores resultados, que foram a utilização dos conjuntos tf-idf sem lematização, tf-idf com lematização e ocorrências com lematização. A Tabela 4.1 mostra os melhores resultados ordenados pela precisão.

Após os primeiros resultados e a seleção dos melhores conjuntos de técni-

Tabela 4.1: Melhores resultados da primeira bateria de testes, em termos de precisão, utilizando anotações de um único anotador e extração de unigramas.

Algoritmo	**Características**	**Média Ponderada**	**Desvio Padrão**
Máxima Entropia	tf-idf/lematização	0,5264	0,0347
Máxima Entropia	ocorrência/lematização	0,5261	0,0356
SVM - Linear	tf-idf	0,5240	0,0341
Máxima Entropia	tf-idf	0,5223	0,0467
Bayes - Multinomial	tf-idf	0,5217	0,0525
SVM - Linear	ocorrência/lematização	0,5212	0,0340
SVM - Linear	tf-idf/lematização	0,5207	0,0203
SVM - Nu	ocorrência/lematização	0,5207	0,0334
SVM - Nu	tf-idf/lematização	0,5163	0,0322
Bayes - Multinomial	ocorrência/lematização	0,5155	0,0272
Bayes - Multinomial	tf-idf/lematização	0,5120	0,0392
Bayes- Bernoulli	tf-idf	0,5083	0,0493
SVM - Nu	tf-idf	0,5068	0,0325
Bayes- Bernoulli	ocorrência/lematização	0,4915	0,0207
Bayes- Bernoulli	tf-idf/lematização	0,4875	0,0343

cas e características, foram realizados novos testes utilizando apenas bigramas, e unigramas em conjunto com bigramas. Os resultados com apenas bigramas foram inferiores até mesmo aos piores resultados com unigramas, porém o uso conjunto representou uma pequena melhora. Os resultados do uso conjunto estão representados na Tabela 4.2.

A terceira e última bateria de testes foi realizada a partir da utilização de trigramas. Foram feitos testes com unigramas e trigramas em conjunto e também com uni, bi e trigramas combinados. A utilização de uni, bi e trigramas em conjunto foi ligeiramente superior à utilização sem bigramas. Os resultados podem ser vistos na Tabela 4.3.

A partir destes testes também foi possível identificar diversos resultados individuais com precisão acima de 70, com um máximo de 75,2 em um único caso.

Tabela 4.2: Melhores resultados da segunda bateria de testes, em termos de precisão, utilizando anotações de um único anotador e extração de unigramas e bigramas em conjunto.

Algoritmo	**Características**	**Média Ponderada**	**Desvio Padrão**
Bayes - Multinomial	tf-idf	0,55076	0,06393
Bayes - Multinomial	tf-idf/lematização	0,54771	0,08290
Máxima Entropia	tf-idf	0,54477	0,02628
Máxima Entropia	tf-idf/lematização	0,54288	0,02650
Máxima Entropia	ocorrência/lematização	0,54178	0,03308
Bayes - Multinomial	ocorrência/lematização	0,52659	0,03681
SVM - Nu	ocorrência/lematização	0,52399	0,04179
SVM - Linear	tf-idf	0,52175	0,03499
SVM - Linear	ocorrência/lematização	0,52142	0,02783
SVM - Linear	tf-idf/lematização	0,51920	0,02284
Bayes - Bernoulli	ocorrência/lematização	0,48914	0,05794
Bayes - Bernoulli	tf-idf/lematização	0,48848	0,05968
Bayes - Bernoulli	tf-idf	0,48434	0,08028
SVM - Nu	tf-idf	0,47719	0,07782
SVM - Nu	tf-idf/lematização	0,47060	0,05541

Tabela 4.3: Melhores resultados da terceira bateria de testes, em termos de precisão, utilizando anotações de um único anotador e extração de unigramas, bigramas e trigramas em conjunto

Algoritmo	**Características**	**Média Ponderada**	**Desvio Padrão**
Máxima Entropia	tf-idf/lematização	0,58938	0,05222
Máxima Entropia	tf-idf	0,56950	0,08045
Bayes - Multinomial	tf-idf/lematização	0,55152	0,06491
Bayes - Multinomial	tf-idf	0,54569	0,08455
Máxima Entropia	ocorrência/lematização	0,53838	0,03015
Bayes - Multinomial	ocorrência/lematização	0,53546	0,02486
SVM - Nu	ocorrência/lematização	0,53093	0,03446
SVM - Linear	tf-idf/lematização	0,52975	0,03570
SVM - Linear	tf-idf	0,52957	0,03072
SVM - Linear	ocorrência/lematização	0,52426	0,03054
SVM - Nu	tf-idf	0,48820	0,06614
SVM - Nu	tf-idf/lematização	0,47987	0,08284
Bayes - Bernoulli	ocorrência/lematização	0,47725	0,05476
Bayes - Bernoulli	tf-idf	0,47212	0,03667
Bayes - Bernoulli	tf-idf/lematização	0,46844	0,06308

No total dos testes, o melhor classificador foi o de máxima entropia com resultados de 58,9, 51,3, e 44,1 de precisão, *recall* e *F-score*, respectivamente.

Partindo da bateria que apresentou os melhores resultados, ao fazer a repetição da bateria de testes com o corpus resultante da concordância dos voluntários, os resultados foram substancialmente melhores em todas as medidas utilizadas. Os melhores resultados dessa nova bateria de testes podem ser vistos, com mais informações, na Tabela 4.4.

Tabela 4.4: Melhores resultados da bateria de testes com o corpus resultante do cruzamento das anotações, com extração unigramas, bigramas e trigramas em conjunto. Os melhores em cada medida estão destacados em negrito.

Classificador	**Extração**	**Precisão**	**Recall**	**F-Score**	**Acurácia**
Máxima Entropia	tf-idf	**0,699** ± 0,0455	0,655 ± 0,0267	0,591 ± 0,0304	0,648 ± 0,0173
SVM Linear	tf-idf	0,693 ± 0,0381	**0,694** ± 0,0243	**0,657** ± 0,0264	0,691 ± 0,0159
Máxima Entropia	Ocorrência	0,673 ± 0,0334	0,689 ± 0,0296	0,646 ± 0,0318	**0,693** ± 0,0346

Da mesma forma que nos primeiros testes, notou-se a existência de vários resultados individuais bastante melhores que a média, com um melhor caso de precisão obtida de 82,2.

Após todos os testes, um experimento foi realizado utilizando *stop words* a partir da lista das 100 palavras mais frequentes do português brasileiro, extraída de *http:www.linguateca.ptchavestopwords*. O resultado do experimento foi três pontos inferior em cada métrica utilizada.

As tabelas com os resultados do experimento estão no Apêndice B.

4.3 Avaliação dos Resultados

Avaliando os resultados, fica evidente a importância da lematização e do uso de tf-idf, uma vez que os melhores resultados observados possuem, ao menos, um desses dois fatores em comum.

A adição de bigramas e trigramas resultou em uma pequena melhora no desempenho quando usadas em conjunto com os unigramas. Essa observação evidencia que a contemplação de expressões de mais de uma palavra só causa ganho de desempenho quando utilizadas, também, as palavras individualmente.

O uso da lematização resulta em melhor desempenho devido ao seu poder de diminuir o número de possíveis expressões (uni, bi e trigramas) ausentes no conjunto de treinamento porém presente no de testes. Essa aplicação tem o efeito colateral de causar a perda eventual do sentido de algumas expressões. Em casos de corpus reduzidos os ganhos superam as possíveis perdas. Já, no caso de um corpus suficientemente grande, o custo-benefício deve ser melhor avaliado.

O desempenho inferior no experimento utilizando *stop words* pode ser explicado devido ao tamanho reduzido do corpus, sendo, portanto, esperado que algumas palavras, mesmo estando entre as mais comuns do português brasileiro, sejam significativas para um sentimento em particular.

A ocorrência de resultados individuais, de uma rodada dentro de cada bateria de teste, que se mostram muito superiores e também muito inferiores aos da média de cada caso, demonstra uma grande dependência do conjunto de treinamento utilizado. Tal dependência deve diminuir em relação diretamente proporcional ao aumento e diversificação do corpus utilizado, na medida em que diminuem as chances de expressões não contempladas no treinamento aparecerem durante os testes. Caso

o mesmo comportamento seja observado em um corpus maior e diversificado, será evidência de necessidade de experimentação com métodos que apresentem variações menores nos resultados.

Ficou clara a importância de um corpus produzido a partir de um estudo de concordância entre anotadores. Os melhores resultados tendo sido mais de 10% superiores aos melhores do corpus sem estudo de concordância consiste em evidência forte o suficiente para confirmar a necessidade de um trabalho de aprimoramento de qualquer corpus que seja proposto à prática de análise de sentimento. Além disso, o valor de 61,3% de concordância entre o anotadores serve como uma referência para comparação do desempenho dos classificadores, visto que os relatos e diferentes resultados encontrados na literatura atestam que o tipo do corpus utilizado possui uma influência que pode ser favorável ou desfavorável ao desempenho do sistema.

Nenhum classificador específico pode ser apontado como melhor, pois, se considerados os desvios padrão, vários equivalem-se como tendo os melhores resultados.

5 CONCLUSÕES E TRABALHOS FUTUROS

Foram avaliadas técnicas e combinações de extração de características para a tarefa de identificação de três emoções e mais um estado neutro em falas transcritas de um filme brasileiro. Utilizando um corpus anotado por dois voluntários, após o cruzamento das anotações, obteve-se um resultante contendo o total de 1.319 frases. O melhor desempenho deu-se com uni, bi e trigramas combinados e também realizando-se lematização. Para cada medida utilizada, um classificador apresentou melhor desempenho, tendo, nos melhores casos, obtido desempenho com precisão de 69,9 em um classificador implementado usando a técnica de máxima entropia e treinado com 70% corpus com a extração de tf-idf. Quanto ao *recall* e *F-score*, os resultados de 69,4 e 65,7, respectivamente, deram-se com o classificador SVM Linear, também utilizando tf-idf e com 90% do corpus para treinamento. Já quanto à acurácia, o melhor foi o classificador de máxima entropia, com 69,3%, treinado com 90% do corpus e utilizando a extração de ocorrências. Comparando com os resultados encontrados na literatura, principalmente com os de (MARTINAZZO, 2010) e (FERREIRA; DOSCIATTI; PARAISO, 2013), pode verificar-se que os resultados são quase equivalentes ou muito superiores, dependendo do método de avaliação escolhido pela autora do trabalho, podendo ser 1,5% inferior ou até 27,85% superior, considerando apenas a acurácia, medida utilizada pelos autores nos dois trabalhos. Quanto ao trabalho de (BELLEGARDA, 2011), os resultados superam quanto à precisão e ao *F-score* e são inferiores quanto ao *recall* e acurácia.

A partir de uma análise dos resultados obtidos foi possível observar o efeito positivo da adição de mais características, utilizando em conjunto os uni, bi e trigramas e também do ganho obtido através da lematização e anotação por mais de um voluntário, que reduziu o número de expressões presentes no conjunto de tes-

tes, mas ausentes no de treinamento e também do número de frases com potencial de interpretação dúbia, causando discordância entre sistemas considerados como as melhores referências para identificação de emoções que é o do próprio ser humano.

Como proposta de trabalhos futuros ficam a expansão do corpus e anotação por mais de duas pessoas, inicialmente com transcrições de filmes de mesmo gênero e em um segundo momento de outros também, além de propostas e testes com extrações de características que tratam o texto além de um simples agrupamento de expressões e suas contagens, mas considerando também outras características do texto, como a identificação de um diálogo e seus atores.

REFERÊNCIAS

AGRAWAL, A.; AN, A. Unsupervised Emotion Detection from Text Using Semantic and Syntactic Relations. **Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01**, Washington, DC, USA, p.346–353, 2012.

BAI, X. Predicting consumer sentiments from online text. **Decision Support Systems**, Amsterdam, The Netherlands, The Netherlands, v.50, n.4, p.732–742, Mar. 2011.

BELLEGARDA, J. R. A Data-Driven Affective Analysis Framework Toward Naturally Expressive Speech Synthesis. **Audio, Speech, and Language Processing, IEEE Transactions on**, [S.l.], v.19, n.5, p.1113–1122, July 2011.

BELLEGARDA, J. R. DATA-DRIVEN ANALYSIS OF EMOTION IN TEXT USING LATENT AFFECTIVE FOLDING AND EMBEDDING. **Computational Intelligence**, [S.l.], v.29, n.3, p.506–526, 2013.

BERGER, A. L.; PIETRA, V. J. D.; PIETRA, S. A. D. A Maximum Entropy Approach to Natural Language Processing. **Comput. Linguist.**, Cambridge, MA, USA, v.22, n.1, p.39–71, Mar. 1996.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural Language Processing with Python**. 1st.ed. Sebastopol, CA, USA: O'Reilly Media, 2009.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, [S.l.], v.20, n.3, p.273–297, 1995.

CRAMMER, K.; SINGER, Y. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. **J. Mach. Learn. Res.**, [S.l.], v.2, p.265–292, Mar. 2002.

CRUZ SILVA, D. da. **Algoritmos de Processamento da Linguagem e Síntese de Voz com Emoções Aplicados a um Conversor Texto-Fala Baseado em HMM**. 2011. Tese (Doutorado em Engenharia Elétrica) — UFRJ/PEE, Rio de Janeiro, RJ, Brasil.

EKMAN, P. Facial Expression and Emotion. **American Psychologist**, [S.l.], v.48, n.4, p.384–392, Apr. 1993.

FAN, R.-E. et al. LIBLINEAR: a library for large linear classification. **The Journal of Machine Learning Research**, [S.l.], v.9, p.1871–1874, 2008.

FERREIRA, L. P. C.; DOSCIATTI, M. M.; PARAISO, E. C. Um Método para o Pré-Processamento de Textos na Identificação Automática de Emoções para o Português do Brasil. **Anais do III Workshop de Iniciação Científica em Tecnologia da Informação e da Linguagem Humana**, [S.l.], p.19–21, Oct. 2013.

GEURTS, P.; ERNST, D.; WEHENKEL, L. Extremely randomized trees. **Machine learning**, [S.l.], v.63, n.1, p.3–42, 2006.

GUYON, I.; BOSER, B.; VAPNIK, V. Automatic capacity tuning of very large VC-dimension classifiers. **Advances in neural information processing systems**, [S.l.], p.147–147, 1993.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. J. H. **The Elements of Statistical Learning**. [S.l.]: Springer New York, 2001. v.1.

KIM, S. M.; VALITUTTI, A. Evaluation of Unsupervised Emotion Models to Textual Affect Recognition. **Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text**, [S.l.], p.62–70, June 2010.

MARTINAZZO, B. **Um Método de Identificação de Emoções em Textos Curtos para o Português do Brasil**. 2010. Tese (Mestrado em Informática) — PUC/PR/PPGIa, Curitiba, PR, Brasil.

MARTÍNEZ-CÁMARA, E. et al. Sentiment analysis in Twitter. **Natural Language Engineering**, [S.l.], v.FirstView, p.1–28, 10 2013.

MCCALLUM, A.; NIGAM, K. et al. A Comparison of Event Models for Naive Bayes Text Classification. **AAAI-98 workshop on learning for text categorization**, [S.l.], v.752, p.41–48, 1998.

METSIS, V.; ANDROUTSOPOULOS, I.; PALIOURAS, G. Spam Filtering with Naive Bayes-Which Naive Bayes? **3rd Conf. on Email and Anti-Spam (CEAS)**, [S.l.], p.27–28, 2006.

NIGAM, K.; LAFFERTY, J.; MCCALLUM, A. Using maximum entropy for text classification. **IJCAI-99 workshop on machine learning for information filtering**, [S.l.], v.1, p.61–67, 1999.

OLSHEN, L. B. J. F. R.; STONE, C. J. Classification and regression trees. **Wadsworth International Group**, [S.l.], 1984.

ORENGO, V.; HUYCK, C. A stemming algorithm for the portuguese language. **String Processing and Information Retrieval, 2001. SPIRE 2001. Proceedings.Eighth International Symposium on**, [S.l.], p.186–193, 2001.

PANG, B.; LEE, L. Opinion Mining and Sentiment Analysis. **Found. Trends Inf. Retr.**, Hanover, MA, USA, v.2, n.1-2, p.1–135, Jan. 2008.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs Up?: sentiment classification using machine learning techniques. **Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10**, Stroudsburg, PA, USA, p.79–86, 2002.

PARDO, T. A. S. et al. Computational linguistics in Brazil: an overview. **Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas**, Stroudsburg, PA, USA, p.1–7, 2010.

PEDREGOSA, F. et al. Scikit-learn: machine learning in python. **Journal of Machine Learning Research**, [S.l.], v.12, p.2825–2830, Nov. 2011.

PRABOWO, R.; THELWALL, M. Sentiment analysis: a combined approach. **Journal of Informetrics**, [S.l.], v.3, n.2, p.143 – 157, 2009.

QUAN, C.; REN, F. Recognizing sentence emotions based on polynomial kernel method using Ren-CECps. **Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference on**, [S.l.], p.1–7, 2009.

QUINLAN, J. R. **C4.5**: programs for machine learning. [S.l.]: Morgan kaufmann, 1993. v.1.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. **Inf. Process. Manage.**, Tarrytown, NY, USA, v.24, n.5, p.513–523, Aug. 1988.

SI, J. et al. Exploiting Topic based Twitter Sentiment for Stock Prediction. **Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics**, [S.l.], p.24–29, Mar. 2013.

STRAPPARAVA, C.; MIHALCEA, R. Learning to identify emotions in text. **Proceedings of the 2008 ACM symposium on Applied computing**, New York, NY, USA, p.1556–1560, 2008.

TRILLA, T.; ALIAS, F. Sentence-Based Sentiment Analysis for Expressive Text-to-Speech. **Audio, Speech, and Language Processing, IEEE Transactions on**, [S.l.], v.21, n.2, p.223–233, 2013.

VAUDABLE, C.; DEVILLERS, L. Negative emotions detection as an indicator of dialogs quality in call centers. **Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on**, [S.l.], p.5109–5112, 2012.

WANG, S. et al. A Feature Selection Method Based on Improved Fisher's Discriminant Ratio for Text Sentiment Classification. **Expert Systems with Applications**, [S.l.], v.38, n.7, p.8696 – 8702, 2011.

ZHANG, H. The Optimality of Naive Bayes. **A A**, [S.l.], v.1, n.2, p.3, 2004.

APÊNDICE A TUTORIAL DO SCIKIT-LEARN

Através do sítio oficial do projeto do *scikit-learn* é possível ter acesso a uma documentação básica adequada, mas apenas em inglês.

Aqui serão dadas instruções rudimentares de operação com o *toolkit* e ao fim será apresentado um exemplo prático,funcional e simples, devidamente explicado.

A.1 Localização, Instalação e Dependências

O sítio oficial do projeto é `http://scikit-learn.org` e através dele é possível ter acesso às instruções de instalação para diversos sistemas operacionais e suas dependências.

Para esta pesquisa foi utilizado o *scikit-learn* versão 0.13 e a versão estável mais atual é a 0.14. Foi utilizado o *Python* versão 2.7.3, que pode ser encontrado, assim como outras versões, no sítio `http://www.python.org`. Fez-se uso da ferramenta *setuptools* na versão 0.6c11 que pode ser encontrada no sítio `https://pypi.python.org/pypi/setuptools`. Esta ferramenta foi a utilizada para realizar o processo de instalação do *Scikit-learn*, facilitando o trabalho de lidar com suas dependências. As bibliotecas *NumPy* e *SciPy* utilizadas foram as de versão, 1.6.2 e 0.9.0, respectivamente. Elas podem ser encontradas nos sítios `http://www.numpy.org` e `http://www.scipy.org`.

A.2 Corpus

Para treinar e testar os classificadores faz-se necessária a utilização de um corpus e de sua anotação. O corpus a ser usado pode resumir-se a um arquivo de texto simples, com conteúdo adequado ao fim que se deseja. Para o caso dessa pesquisa, o corpus criado consistiu em um arquivo com uma frase por linha.

A.3 Treinamento dos Classificadores

O *toolkit* disponibiliza uma série de classificadores, de diversos tipos e para diversas aplicações. Os utilizados neste trabalho, que implementam os algoritmos NB, ME e SVM, servem para o propósito deste trabalho e tem uma operação básica semelhante.

É necessário que o corpus seja previamente transformado segundo a extração de características escolhida. Uma abordagem, das mais simples, é utilizar unigramas, o conceito de *bag of words* e considerar apenas ocorrências. Dessa forma, cada elemento que se pretende classificar pode ser transformado em um dicionário, estrutura de dados nativa da linguagem *Python*, onde cada unigrama seria uma chave e teria como valor 0 ou 1 para indicar sua ocorrência.

O conjunto de treinamento deverá consistir de duas listas, onde a primeira deverá ser, segundo nossa exemplificação, uma lista de dicionários e a segunda, uma lista de rótulos dados a esses elementos na fase de anotação. Esse par de listas, representa, então, todo o corpus e suas anotações.

Para realizar o treinamento do classificador, para o caso desta exemplificação,

é necessária mais uma etapa intermediária. É necessário transformar a lista de dicionários únicos em um vetor numérico de tamanho fixo que seja capaz de representar cada um dos elementos traduzidos em forma de dicionário. O próprio *toolkit* oferece uma ferramenta para este fim.

Após essa transformação, as duas listas poderão ser passadas à função de treinamento do classificador escolhido.

Detalhes práticos em A.5.

A.4 Utilização dos Classificadores

Após serem devidamente treinados, os classificadores serão capazes de classificar os elementos que se desejam avaliar. O conjunto que se deseja avaliar deve ser transformado da mesma maneira que o conjunto de treinamento. É uma boa prática transformar todo o corpus conjuntamente e, após a transformação, dividi-lo em conjuntos de teste e treinamento.

Os elementos podem ser classificados, sendo passados, em forma de lista, à função de classificação do classificador treinado. O retorno da função consiste em uma lista com os rótulos referentes a classificação efetuada. A ordem da lista, como esperado, corresponde à ordem dos elementos

Detalhes práticos em A.5.

A.5 Exemplo Simples

Aqui é demonstrado um exemplo de utilização simples do *toolkit*. O exemplo consiste no fluxo normal e esperado da aplicação das ferramentas já, brevemente, descritas aqui. O código mostra a aplicação de um classificador do tipo SVM a um corpus de apenas 5 frases, com o propósito de classificação do sentimento das frases entre *positivo* e *negativo*. O treinamento será feito com 4 frases e o teste com apenas uma. O código do exemplo foi testado é funcional e está comentado descrevendo a função de cada linha.

Segue o código de exemplo:

```
1  # Importa classificador LinearSVC.
2  # Este implementa um classificador SVM com funcao de kernel
       do tipo linear.
3  # Os outros classificadores utilizados na pesquisa funcionam
       de forma analoga.
4
5  from sklearn.svm import LinearSVC
6
7  # Importa o elemento que transforma uma lista de dicionarios
       em uma lista de vetores numericos.
8
9  from sklearn.feature_extraction import DictVectorizer
10
11 # Lista pequena que simula um corpus.
12
13 lista_frases = ['Filme bom!',
14                  'Filme ruim!',
15                  'Filme muito bom!',
16                  'Filme muito ruim!',
17                  'Bom filme!'
18                  ]
19
20 # Lista de dicionarios, cada um representando as frases do
       corpus.
```

```
21
22 dicionario = [{'filme':1, 'bom':1},
23                {'filme':1, 'ruim':1},
24                {'filme':1, 'muito':1, 'bom':1},
25                {'filme':1, 'muito':1, 'ruim':1},
26                {'bom':1, 'filme':1}
27                ]
28
29 # Lista com as anotacoes referentes as frases do corpus.
30
31 lista_anotacoes = ['positivo',
32                     'negativo',
33                     'positivo',
34                     'negativo',
35                     'positivo'
36                     ]
37
38 # Instanciacao do transformador.
39
40 transformador = DictVectorizer()
41
42 # Funcao do transformador que traduz a lista de dicionarios
      em uma lista de vetores numericos, propria para
      treinamento e teste do classificador.
43
44 lista_transformada = transformador.fit_transform(dicionario)
      .toarray()
45
46 # Instanciacao do classificador LinearSVC, com seus
      parametros padrao.
47
48 classificador = LinearSVC()
49
50 # Treinamento do classificador.
51 # Ele recebe como parametros duas listas, a primeira, com os
       vetores para treinamento e a segunda, com as anotacoes.
52 # Ambas as listas sao passadas com todos seus elementos,
      menos o ultimo, deixado para ser usado como teste.
53
54 classificador.fit(lista_transformada[:-1], lista_anotacoes
```

```
    [:-1])
55
56 # Avalia a lista passada para classificacao.
57 # A lista possui apenas um elemento, representando a ultima
       frase do corpus.
58
59 resultado = classificador.predict(lista_transformada[-1:])
60
61 # Imprime lista com o resultado da classificacao.
62
63 print resultado
64
65 # Imprime resultado da transformacao da lista de dicionarios
       .
66
67 print lista_transformada
68
69 # Imprime a transformacao inversa da lista de vetores
       numericos.
70
71 print transformador.inverse_transform(lista_transformada)
```

Ao executar esse código, é esperado que sejam impressas três respostas.

A linha de código 63 deve mostrar como resultado a resposta do classificador, que se espera ser:

['positivo'].

A linha de código 67 deve mostrar como resultado a lista dos vetores numéricos, criados a partir da lista de dicionários. A partir da observação do que é impresso como resultado dessa linha de código e relacionando com as frases do corpus, é possível visualizar e entender a representação de cada posição do vetor com cada palavra de cada frase. Vale ressaltar que a ordem em que as palavras aparecem

na frase não é obedecida, apenas a ocorrência é considerada. A impressão esperada é:

```
[[ 1.   1.   0.   0.]
 [ 0.   1.   0.   1.]
 [ 1.   1.   1.   0.]
 [ 0.   1.   1.   1.]
 [ 1.   1.   0.   0.]].
```

A linha de código 71 deve mostrar como resultado, a lista de dicionários, recriada a partir dos vetores numéricos. Isso serve como método para verificar se a transformação ocorreu corretamente, respeitando os dicionários. A impressão esperada é[1]:

```
[{'bom': 1.0, 'filme': 1.0},
{'ruim': 1.0, 'filme': 1.0},
{'muito': 1.0, 'bom': 1.0, 'filme': 1.0},
{'muito': 1.0, 'filme': 1.0, 'ruim': 1.0},
{'bom': 1.0, 'filme': 1.0}].
```

[1]A impressão mostrada coloca cada item da lista em uma linha diferente. Isso foi feito para melhor exibição neste texto. A impressão realizada pelo programa deve ser em apenas uma linha.

APÊNDICE B TABELAS DE RESULTADOS

A seguir estão dispostas as tabelas com os resultados dos testes realizados.

B.1 Resultados

Tabela B.1: Melhores resultados da primeira bateria de testes com o corpus com anotações do anotador 1 e extração de unigramas e ocorrências; sem lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.49641	0.51198	0.45539
Bayes Multinomial	0.50249	0.51874	0.49081
Extremely Randomized Trees	0.45142	0.46298	0.44954
Árvore de Decisão	0.46216	0.47127	0.45853
LDA	0.46389	0.46897	0.46378
Máxima Entropia	0.51836	0.53094	0.51118
SVM Linear	0.50827	0.51966	0.50746
SVM	0.13591	0.36866	0.19861
SVM Nu	0.49962	0.51142	0.49193

Tabela B.2: Melhores resultados da primeira bateria de testes com o corpus com anotações do anotador 1 e extração de unigramas e ocorrências; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.49145	0.51196	0.46563
Bayes Multinomial	0.51551	0.52842	0.50411
Extremely Randomized Trees	0.45532	0.46498	0.45295
Árvore de Decisão	0.46006	0.47097	0.46013
LDA	0.48384	0.49017	0.48383
Máxima Entropia	0.5261	0.53502	0.51659
SVM Linear	0.52123	0.52903	0.51964
SVM	0.13591	0.36866	0.19861
SVM Nu	0.52068	0.52611	0.50876

Tabela B.3: Melhores resultados da primeira bateria de testes com o corpus com anotações do anotador 1 e extração de unigramas e frequências; sem lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.49246	0.50753	0.45115
Bayes Multinomial	0.50366	0.52089	0.49435
Extremely Randomized Trees	0.44917	0.45853	0.44754
Árvore de Decisão	0.44555	0.45486	0.44159
LDA	0.44917	0.4533	0.44797
Máxima Entropia	0.52589	0.5318	0.51267
SVM Linear	0.50837	0.51674	0.50647
SVM	0.13591	0.36866	0.19861
SVM Nu	0.50828	0.51459	0.49712

Tabela B.4: Melhores resultados da primeira bateria de testes com o corpus com anotações do anotador 1 e extração de unigramas e frequências; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.42792	0.42135	0.33086
Bayes Multinomial	0.44904	0.43656	0.36593
Extremely Randomized Trees	0.38358	0.39401	0.38005
Árvore de Decisão	0.38175	0.39324	0.37836
LDA	0.43277	0.44071	0.41579
Máxima Entropia	0.42856	0.43948	0.40473
SVM Linear	0.42837	0.43963	0.41128
SVM	0.13591	0.36866	0.19861
SVM Nu	0.40459	0.40968	0.39157

Tabela B.5: Melhores resultados da primeira bateria de testes com o corpus com anotações do anotador 1 e extração de unigramas e tf-idf; sem lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.50825	0.51874	0.46345
Bayes Multinomial	0.52173	0.50753	0.42665
Extremely Randomized Trees	0.4433	0.44992	0.43941
Árvore de Decisão	0.44015	0.44668	0.43901
LDA	0.46886	0.47634	0.46948
Máxima Entropia	0.52232	0.52166	0.47755
SVM Linear	0.52397	0.53287	0.52125
SVM	0.13591	0.36866	0.19861
SVM Nu	0.50678	0.51075	0.50438

Tabela B.6: Melhores resultados da primeira bateria de testes com o corpus com anotações do anotador 1 e extração de unigramas e tf-idf; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.48754	0.51026	0.46339
Bayes Multinomial	0.51203	0.50995	0.43677
Extremely Randomized Trees	0.4453	0.45401	0.44468
Árvore de Decisão	0.45529	0.46006	0.45349
LDA	0.48556	0.48909	0.48512
Máxima Entropia	0.52636	0.52535	0.48767
SVM Linear	0.52072	0.5314	0.52056
SVM	0.13591	0.36866	0.19861
SVM Nu	0.51626	0.52028	0.51302

Tabela B.7: Melhores resultados da segunda bateria de testes com o corpus com anotações do anotador 1 e extração de bigramas e ocorrências; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.4579	0.42657	0.31746
Bayes Multinomial	0.46556	0.48387	0.4592
Máxima Entropia	0.50399	0.48525	0.43472
SVM Linear	0.48079	0.48172	0.45663
SVM Nu	0.49626	0.48648	0.44179

Tabela B.8: Melhores resultados da segunda bateria de testes com o corpus com anotações do anotador 1 e extração de bigramas e tf-idf; sem lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.46029	0.42442	0.31212
Bayes Multinomial	0.53743	0.47435	0.3863
Máxima Entropia	0.5353	0.46636	0.37579
SVM Linear	0.50174	0.48341	0.44746
SVM Nu	0.45556	0.38372	0.36315

Tabela B.9: Melhores resultados da segunda bateria de testes com o corpus com anotações do anotador 1 e extração de bigramas e tf-idf; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.46157	0.43134	0.32526
Bayes Multinomial	0.53619	0.47819	0.39622
Máxima Entropia	0.53852	0.47189	0.3857
SVM Linear	0.48743	0.47742	0.44361
SVM Nu	0.457	0.39109	0.37654

Tabela B.10: Melhores resultados da terceira bateria de testes com o corpus com anotações do anotador 1 e extração de uni e bigramas e ocorrências; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.48914	0.47143	0.38385
Bayes Multinomial	0.52659	0.53779	0.51541
Máxima Entropia	0.54178	0.54777	0.52643
SVM Linear	0.52142	0.52886	0.52036
SVM Nu	0.52399	0.53057	0.50387

Tabela B.11: Melhores resultados da terceira bateria de testes com o corpus com anotações do anotador 1 e extração de uni e bigramas e tf-idf; sem lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.48434	0.46528	0.37491
Bayes Multinomial	0.55076	0.49862	0.40727
Máxima Entropia	0.54477	0.50799	0.44075
SVM Linear	0.52175	0.52934	0.51223
SVM Nu	0.47719	0.42446	0.41331

Tabela B.12: Melhores resultados da terceira bateria de testes com o corpus com anotações do anotador 1 e extração de uni e bigramas e tf-idf; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.48848	0.47083	0.383
Bayes Multinomial	0.54771	0.5034	0.41584
Máxima Entropia	0.54288	0.51429	0.45299
SVM Linear	0.5192	0.52739	0.51209
SVM Nu	0.4706	0.43248	0.42476

Tabela B.13: Melhores resultados da quarta bateria de testes com o corpus com anotações do anotador 1 e extração de uni e trigramas e ocorrências; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.49703	0.4573	0.36081
Bayes Multinomial	0.5214	0.53057	0.49893
Máxima Entropia	0.53852	0.54424	0.52443
SVM Linear	0.51894	0.5298	0.51685
SVM Nu	0.52164	0.52693	0.50277

Tabela B.14: Melhores resultados da quarta bateria de testes com o corpus com anotações do anotador 1 e extração de uni e trigramas e tf-idf; sem lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.49376	0.44946	0.35142
Bayes Multinomial	0.55996	0.49447	0.40023
Máxima Entropia	0.53947	0.50691	0.4359
SVM Linear	0.52412	0.53118	0.51088
SVM Nu	0.46687	0.42029	0.40846

Tabela B.15: Melhores resultados da quarta bateria de testes com o corpus com anotações do anotador 1 e extração de uni e trigramas e tf-idf; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.49812	0.45315	0.35507
Bayes Multinomial	0.51581	0.50023	0.40931
Máxima Entropia	0.54146	0.51397	0.44947
SVM Linear	0.53582	0.54086	0.52489
SVM Nu	0.47566	0.43218	0.42174

Tabela B.16: Melhores resultados da quinta bateria de testes com o corpus com anotações do anotador 1 e extração de uni, bi e trigramas e ocorrências; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.47725	0.45253	0.35141
Bayes Multinomial	0.53546	0.54547	0.52833
Máxima Entropia	0.53838	0.54178	0.52118
SVM Linear	0.52426	0.5321	0.52106
SVM Nu	0.53093	0.52919	0.50308

Tabela B.17: Melhores resultados da quinta bateria de testes com o corpus com anotações do anotador 1 e extração de uni, bi e trigramas e tf-idf; sem lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.47212	0.43963	0.33256
Bayes Multinomial	0.54569	0.50169	0.41044
Máxima Entropia	0.5695	0.5086	0.43411
SVM Linear	0.52957	0.53088	0.50917
SVM Nu	0.4882	0.41566	0.38861

Tabela B.18: Melhores resultados da quinta bateria de testes com o corpus com anotações do anotador 1 e extração de uni, bi e trigramas e tf-idf; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**
Bayes Bernoulli	0.46844	0.44378	0.34177
Bayes Multinomial	0.55152	0.50384	0.41199
Máxima Entropia	0.58938	0.51567	0.44398
SVM Linear	0.52975	0.53518	0.51566
SVM Nu	0.47987	0.42276	0.40031

Tabela B.19: Melhores resultados da sexta bateria de testes com o corpus com anotações em comum dos dois anotadores e extração de uni, bi e trigramas e ocorrências; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**	**Acurácia**
Bayes Multinomial	0.63821	0.66642	0.64593	0.66241
Máxima Entropia	0.67293	0.68905	0.6463	0.69279
SVM Linear	0.64585	0.67015	0.64752	0.67068

Tabela B.20: Melhores resultados da sexta bateria de testes com o corpus com anotações em comum dos dois anotadores e extração de uni, bi e trigramas e tf-idf; com lematização.

Classificador	**Precisão**	**Recall**	**F-Score**	**Acurácia**
Bayes Multinomial	0.68468	0.66144	0.59755	0.65627
Máxima Entropia	0.69924	0.65522	0.59072	0.64787
SVM Linear	0.69274	0.69403	0.65681	0.69098