Analise de Sentimento polarizada

Raissa Camelo Salhab

[1]Departamento de Computação – Universidade Federal Rural Pernambucana (UFRPE)
Bacharelado em Ciência da Computação

srtacamelo@gmail.com

Resumo. *Neste artigo foi realizado um estudo comparativo entre diferentes classificadores para a analise de sentimento de revisões de filmes da base IMDB. As técnicas de pré-processamento utilizadas para lidar com a informação contida em cada arquivo foram descritas e visaram entrar em maior compatibilidade possível com os algoritmos selecionados. O objetivo principal deste documento é estabelecer relações entre os métodos utilizados e os resultados obtidos, afim de destacar quais abordagens funcionam melhor com o modelo proposto. Esse estudo serve como incentivo e um vislumbre do que pode ser feito com a analise de sentimentos, permitindo o leitor a compreender melhor o problema e as soluções disponíveis.*

1. Introdução

A Mineração de textos se trata da extração de características de documentos escritos, afim de reter algum tipo de informação contido neles. Sendo uma das áreas da ciência de dados, a mineração de textos se preocupa em linkar e classificar conteúdos diferentes e tentar achar padrões para os dados selecionados. [Hearst 2003]

Os computadores são extremamente eficientes em organizar e processar dados numéricos e estruturas de dados computacionais, já que já estão representadas de forma inteligível para o mesmo. Contudo, se tratando de textos escritos em língua humana, as máquinas não são capazes de entender e interpretar com a mesma facilidade que os próprios seres humanos. Se faz necessário algum tipo de mecanismo, do qual textos possam ser processados e informações numéricas e inteligíveis pelo computador possam ser extraídas dos documentos.

Através de técnicas linguísticas como lemmanização (extração de radicais), tokenização (extração de palavras e símbolos), Sentence Spliting (divisão em frases), uso de n-grams e cálculos de frequência de termo a mineração de texto vem possibilitado a interpretação e o processamento de dados textuais por computadores. [Muhr 1991]

A categorização de textos,o processamento de linguagem natural, Analise de sentimentos é uma das subáreas da mineração de texto, que consiste em classificar se determinado texto contém opiniões positivas, neutras ou negativas, sobre dado contexto. [Leung 2009]

A partir dessa classificação inúmeros tipos de dados podem ser analisados automaticamente, retirando a necessidade de verificar manualmente cada documento um por um. Os processos de analise de sentimentos geralmente envolvem algoritmos de classificação como o KNN ou até mesmo redes neurais, geralmente optando-se por modelos probabilísticos. [Aggarwal and Zhai 2012] Empresas de serviços ao consumidor têm adotado a analise de sentimento como forma de avaliar a aceitação de seus produtos no mercado, servindo como pesquisa de opinião e possibilitando as empresas a adequarem seus produtos de maneira mais eficiente. [Ravi and Ravi 2015] A analise de sentimentos permite não apenas a classificação de de opiniões de usuários sobre produtos como também

tem sido usada como forma de avaliar popularidade de candidatos a cargos políticos em plataformas sociais, avaliação de sentimentos de usuários de redes sociais em relação a determinado assunto ou noticia. [Pozzi et al. 2016][Godbole et al. 2007] Essa área possui um grande potencial a ser explorado, a partir de sistemas de definição de sentimentos, decisões poderão ser tomadas e até mesmo um algoritmo pode determinar que tipo de propaganda exibir para determinado público, quais produtos sugerir, a ideia é cada vez mais a computação torne a experiencia do usuário personalizável e única para cada individuo.

Neste projeto utilizamos um banco de dados de revisões de filmes da IMDB, afim de classificar as revisões quanto ao seu sentimento. Para a avaliação das classes utilizamos algoritmos distintos, afim de distinguir os textos entre duas classes: opiniões positivas ou negativas. Foi realizada uma comparação de resultados utilizando classificadores diferentes, o objetivo é determinar qual ou quais métodos melhor se adaptam ao problema proposto e por que.

2. Materiais

O conjunto de dados utilizado foi retirado do dataSet da IMDB, que contem mais de 24.000 exemplares de revisão de filmes. A IMDB (Internet Movie database) se trata de um banco de dados gerenciado atualmente pela Amazon, reunindo várias informações sobre programas de TV, músicas, jogos, livros e produtos em geral.
O conjunto escolhido foi um set de textos escritos na língua inglesa contendo opiniões de pessoas sobre diferentes filmes. Cada opinião se trata de um texto de aproximadamente 5 frases, todos salvos no formato *.txt*. São 12.500 arquivos *txt* contendo opiniões negativas sobre filmes e outros 12.500 contendo opiniões positivas, já separados em pastas diferentes. A base é de domínio publico e pode ser baixada através do site da IMDB.
Para o pré-processamento dos dados utilizou-se a biblioteca Python NLTK, de mineração de texto em conjunto com a biblioteca Scikit learn.

3. Pré-processamento

Após adquirida a base de dados alguns procedimentos foram realizados afim de maximizar o potencial dos classificadores e diminuir o volume dos dados, retendo apenas as informações mais relevantes dos documentos. Através de funções de tokenização e remoção de palavras da biblioteca NLTK, cada documento foi convertido em uma lista de tokens (palavras, sinais, símbolos, siglas). Após a tokenização de cada documento individualmente, a lista tokenizada foi utilizada para remoção de *Stop Words*, palavras repetitivas que não agregam significância à classificação, como pronomes, artigos e preposições. Sinais e números também foram removidos de cada documento. Dada a a remoção de palavras e consequente diminuição do tamanho das listas de tokens gerada por cada documento, o ultimo passo do processamento consistiu em extrair valores numéricos do texto resultante, para poder alimentar os classificadores.
Algoritmos como a árvore de decisão poderiam usar a base bruta em formato de texto, porém os demais classificadores abordados (SVM, KNN, etc) necessitam de valores numéricos, por tanto, para fins de comparação justa entre os métodos, a base foi aplicada na sua forma numérica para todos os algoritmos. Para a extração de valores numéricos dos documentos os conceitos de frequência de termo (TF) e frequência inversa de documento (IDF) foram aplicados. O TF computa a frequência de cada termo em determinado documento, expressado pela formula:

Figura 1. Calculo TF

$$TF = nt/n$$

Onde nt é o número de vezes que o termo aparece em um único documento e n é a quantidade total de termos no documento. O IDF determina a frequência inversa que um termo aparece em todo o Corpus, dado por:

Figura 2. Calculo IDF

$$\log(N/dtf)$$

Onde N é a quantidade total de documentos no Corpus e dtf é a quantidade de documentos em que o termo aparece. Os valores de TF e IDF são calculados para todos os termos presentes no corpus. O objetivo do TF é permitir que os classificadores façam relação entre a frequência de determinados termos com a classificação do documento, por exemplo, termos como ”Good”, ”Awesome”e ”Great”tendem a aparecer com mais frequência em documentos de polaridade positiva, enquanto termos como ”Bad”,”Awful”e ”Boring”tendem a aparecer em documentos de polaridade negativa. Porém o uso do TF isoladamente pode fazer com que termos comuns nas duas polaridades assumam valores altos, dado que sua frequência é alta nos dois tipos de documentos, fazendo com que esses termos tenham uma relevância maior para o classificador. Para evitar este problema, o valor do IDF é utilizado como normalizador, pois ao multiplicar o TF com o IDF, documentos que possuem uma frequência muito grande entre documentos distintos tem seu valor diminuído.

Os calculo do TF-IDF foi realizado utilizando as funções da biblioteca sklearn, gerando uma tabela de TF-IDFs onde cada linha da tabela contém os valores de TF-IDF para cada termo em cada documento. A tabela possui um tamanho de d x tn, sendo d a quantidade de documentos no corpus e tn a quantidade total de termos no corpus. A tabela gerada foi salva em formato *.csv*, através da biblioteca pandas, para ser usada posteriormente nos classificadores.

Antes de salvar a tabela tf-idf em formato csv, o valor referente a classificação de cada documento foi adicionado à tabela, (0 para negativo e 1 para positivo). A classificação dos documentos foi conferida através da ordem de leitura dos mesmos antes do pré-processamento. Na base original os arquivos com documentos positivos estão separados em uma pasta e os negativos em outra, apesar da dimensão da base ter sido diminuída, a organização foi mantida, facilitando a caracterização de cada documento. Os primeiros 550 documentos lidos eram positivos, enquanto os 550 seguintes eram negativos, assim só foi necessária a indexação dos valores (0 e 1) nesta ordem, após a concatenação das duas listas de documento e computação dos valores tf-idf.

O passo de calcular os valores tf-idf de cada termo/documento somente após a concatenação das listas de documentos positivos e negativos foi extremamente crucial para a integridade dos dados. O IDF, como visto acima, depende da quantidade total de documentos e de termos do corpus. Caso o tf-idf tivesse sido calculado separadamente para para cada conjunto de dados, os valores seriam significantemente diferentes do que para o conjunto total. Sem contar que a tabela final de valores seria inutilizável para classificação, uma vez que dificilmente a quantidade de termos seria igual para os dois polos de arquivos, gerando uma matriz não quadrada. E, mesmo que por coincidência a matriz resultante fosse quadrada, os dados não seriam corretos, já que para cada polo de arquivos termos diferentes foram levados em consideração, e sua frequência tf-idf estaria enviesada, dado que a quantidade de documentos utilizada para calcular o idf não iria condiz com a real quantidade total de documentos no corpus.

Como o dataset da IMDB é muito extenso, o processamento e a classificação dos dados tomariam muito tempo para serem concluídos. Para evitar uma longa espera, dos 12.500 documentos de cada polaridade apenas 550 de cada foram selecionados para o pré-processamento e posterior uso. Ao utilizar a base inteira na primeira vez que o processo foi realizado, o formato *csv* não foi capaz de suportar o tamanho da tabela gerada e o arquivo final não pôde ser salvo.

4. Metodologia

Para a avaliação dos resultados o método da validação cruzada foi utilizado sobre os dados, através da função k-fold da biblioteca scikit learn. A tabela gerada durante o pre-processamento foi carregada sobre a forma de um dataframe, da biblioteca pandas, para melhor organizar os dados e fornecer os valores de maneira adequada para os classificadores. Após lidos os valores foram randomizados, usando uma função da própria biblioteca pandas que mistura as linhas dos dataframes sem que os valores das colunas se misturem. Após randomizados, os dados foram divididos entre frequências e classes, dividindo-os em dois dataframes separados. A função de k-fold do sklearn ficou responsável por dividir os dados em 10 folds separados e fazer sua iteração num for. a cada iteração do 10-folds o mesmo conjunto de treino e testes foi usado para todos os algoritmos, gerando uma medida de acurácia para todos eles. O método 10-folds foi repetido 5 vezes diferentes, cada vez randomizando novamente o conjunto de dados, gerando no total 50 valores de acurácia para cada classificador. A média, a mediana e o desvio padrão de cada conjunto de acurácia foi extraída e a hipótese de t-studant foi utilizada para avaliar as diferenças entre abordagens modificadas dos mesmos classificadores.

5. Modelos Apresentados

Dentre os classificadores utilizados para a comparação se encontram a árvore de decisão, naive bayes, SVM, KNN e regressão logistica. A seguir:

5.0.1. Árvore de Decisão

A árvore de decisão é um modelo de aprendizagem que representa uma série de ”regras”para se determinar o pertencimento de um dado a determinada classe. Basicamente, as árvores de decisão traçam caminhos com base nas características de cada dado, inferindo a partir destas características a classe deste dado. O modelo de árvore escolhido usa

o algoritmo padrão com base nos valores de entropia de cada característica. Ou seja, a árvore é montada através dos dados treinamento, analisando as características que mais distinguem uma classe de outra primeiro e assim decidindo a que classe o dado pertence.

5.0.2. Naive Bayes

O naive bayes utiliza os valores de treino para criar um modelo probabilístico, presumindo que cada um dos dados seja independente. Desta forma a polaridade de um documento é definida dada a maior tendencia de classe dos dados. Se a maioria dos termos de um documento tendem a ser positivos, logo o documento inteiro é classificado como positivo.

5.0.3. Support Vector Machine

A maquina de vetores suportes cria um modelo não probabilístico de classificação com base nos valores de treinamento, traçando uma reta imaginaria, que dividiria os dados entre as duas polaridades possíveis. Se representarmos os dados em um plano cartesiano, esta reta imaginaria consiste em nada mais nada menos do que uma função que corta o plano, separando as duas classes. Para os testes realizados utilizamos as variações ”RBF”e ”Linear”da SVM.

5.0.4. K- Nearest Neighbours

O metodo KNN calcula a distancia de cada dado do conjunto de testes para cada dado do conjunto de treino (classificado), os k dados do conjunto de treino mais próximos do dado que se deseja classificar são escolhidos e a classe que mais se repetir entre eles é dada ao novo dado. Esse método possui duas variações distintas que foram testadas, o padrão k-nn e o wKNN ou KNN ponderado, que utiliza uma função de distancia para recalcular as distancias entre os dados, fazendo com que os dados mais próximos se aproximem mais e os mais distantes se afastem. [SANTOS et al. 2009]

5.0.5. Regressão Logistica

A regressão logística é uma técnica estatística que visa estimar a probabilidade de ocorrência de determinado ”acontecimento”dada a existência de determinadas características independentes, assim criando um modelo, geralmente dicótomo, de classificação com base nessas características. No problema proposto a regressão logística deve classificar cada texto com base nas frequências de cada termo, ou seja, o algoritmo encontrará padrões de associação entre um determinado conjunto de termos e as classes positiva e negativa.

6. Resultados Esperados

Dada a natureza do problema estima-se que as abordagens probabilísticas como o Naive Bayes e a regressão logística obtenham melhores resultados, dado que cada termo de um

texto carrega uma polaridade individual que por si só já indica a polaridade do texto inteiro. As técnicas de KNN também possuem grandes chances de estarem entre os melhores resultados, pois a distancia entre documentos que possuem frequências semelhantes dos mesmos termos será menor e estima-se que documentos de mesma polaridade contenham termos comuns entre si.

7. Analise Experimental

A analise experimental consistiu na parte mais importante do projeto, promovendo uma comparação entre as distintas abordagens de classificação fornecidas pela biblioteca sklearn. O objetivo final desta sessão é determinar quais técnicas são mais adequadas para o problema proposto e o motivo das mesmas terem obtido um resultado melhor em detrimento das restantes.

Para a analise ser possível foram utilizadas as métricas de acurácia para medir a taxa de acerto de cada classificador, sendo extraídas logo apos os testes com cada algoritmo. As medidas de média, mediana e desvio padrão foram tiradas sobre as 50 acurácias geradas pelas 5 iterações de 10-folds. A biblioteca sklearn providenciou as funções para extração de tais valores.

Os valores da hipotese de t-studant calculados A analise experimental foi dividida em três partes, primeiramente a média, a mediana e o desvio padrão do conjunto de acurácia de cada classificador foi avaliado, e uma comparação entre os resultados foi realizada. O objetivo da primeira analise é, tendo em vista a média de acurácias, destacar quais foram as melhores abordagens, buscando entender e explanar os motivos de tais resultados. Logo em seguida uma comparação entre cada variação dos algoritmos KNN e SVM foi feita baseada na hipótese de t-studant, gerada entre o conjunto de acurácias das variações. Aqui buscamos determinar se houve realmente alguma diferença significante entre as variações, comparando também o resultado da hipótese com as médias de acurácia e verificando a congruência dos dois dados.

Na terceira parte busca-se fazer uma comparação mais a fundo dos métodos cujas médias de acurácia foram maiores, afim de determinar qual das abordagens é mais viável para o problema de polarização de textos. A seguir uma tabela contendo informações sobre a média, mediana e desvio padrão do conjunto de 50 acurácias gerado por cada classificador.

Figura 3. Resultados

Média, Desvio Padrão e Mediana			
Classificador	Média	Desvio Padrão	Mediana
Árvore	0.6536	0.0388	0.6545
Naive Bayes	0.8727	0.0353	0.8636
KNN (n=1)	0.7254	0.0283	0.7363
KNN (n=3)	0.6927	0.0412	0.7090
KNN (n=5)	0.7136	0.0579	0.6954
WKNN (n=1)	0.7254	0.0283	0.7363
WKNN (n=3)	0.6927	0.0412	0.7090
WKNN (n=5)	0.7136	0.0579	0.6954
SVM-LINEAR	0.4763	0.0179	0.4863
SVM-RBF	0.4763	0.0179	0.4863
REG-LOGI	0.8263	0.0622	0.8363

Como o esperado, o uso de técnicas probabilísticas se mostram promissoras na classificação dos textos com base em frequência de termos. O pré-processamento utilizado influencia no resultado, uma vez que modelos probabilísticos retem melhor os padrões em dados baseados em frequência ou ausencia/presença de características. O Naive Bayes obteve uma média de acurácias de aproximadamente 0.65, com um desvio padrão menor que 0.1, indicando uma consistencia dos valores de acurácia na distribuição. Entretanto o Naive Bayes ganha para a regressão logística, que obteve o segundo melhor resultado entre todos os classificadores testados, com uma média de 0.82 de acurácia, também consistente dado o desvio padrão de 0.06. Os classificadores de distãncia (KNN e wKNN) obtiveram valores semelhantes de média, também tendo um bom desempenho. Os algoritmos de distancia geralmente lidam bem com esse tipo de problema, o volume extenso de características por dado não atrapalha a eficiência do KNN, em termos de acurácia, provendo mais informações para distanciar os dados de classes distintas. Surpreendentemente o Naive Bayes obteve uma taxa mais alta que a regressão logística, provavelmente devido a sua característica de observar a penas as probabilidades em relação a cada característica (palavra) isoladamente. A regressão logística poderia ter apresentado um valor melhor, talvez dado o modelo pirobalístico utilizado *lbfg* não seja o melhor para essa abordagem, abrindo espaço para futuros testes. Outra surpresa foi o algoritmo da árvore de decisão, o qual se esperava resultados melhores. A árvore de decisão consegue abstrair quais atributos possuem um grau maior de ”separação”entre classes, criando assim uma série de regras que em uma ordem especifica conseguem distinguir um dado de uma classe de outra, contudo o resultado obtido foi um pouco menor do que os do KNN e da regressão logística, esperava-se que fosse mais próximo.

A seguir encontra-se uma analise entre as variações de KNN e SVM, dada a hipotese de t-studant fornecida na tabela abaixo:

Figura 4. Comparação das variações

KNN e SVM variações		
Comparação	t-value	p-value
KNN-1 e KNN-3	35.345	0.00062
KNN-1 e KNN-5	-0.1146	0.9089
KNN-3 e KNN-5	-30.908	0.0025
WKNN-1 e WKNN-3	35.345	0.00062
WKNN-1 e WKNN-5	-0.1146	0.9089
WKNN-3 e WKNN-5	-3.090	0.0025
SVM-LIN e SVM-RBF	0.0	1.0

Como se pode perceber, as variações entre SVM-LIN e SVM-RBF são praticamente nulas de acordo com a hipótese estatística, não possuindo nenhuma diferença significativa. Entre o KNN (1-NN) e o KNN (3-NN) contudo é apontada uma diferença muito grande entre os dois metódos, se observado novamente na tabela de acurácias o 1NN realmente obteve uma métrica 0.01 maior que a do 3NN, a mesma medida é mostrada entre o W1NN e o W3NN mostrando haver ganho significativo do w1NN em relação ao w3NN. O restante dos classificadores comparados demonstram não ter grande variação entre si, dada a hipótese.
Para terminar nossa analise comparamos agora os três classificadores que obtiveram os melhores resultados: Regressão Logística, 1NN e o Naive Bayes, através da hipótese de t-studant:

Figura 5. Comparação Melhores Classificadores

Melhores Algoritmos		
Comparação	t-value	p-value
KNN-1 e NBAYES	-28.155	9.170
KNN-1 e REG - LOG	-18.904	18.445
NBAYES e REG - LOG	57.047	12.378

Como é observado, não houve grandes ganhos em relação ao KNN e o naive bayes nem ao KNN e a regressão logística. Porém é apontado um ganho significativo da regressão em relação ao Naive Bayes.

8. Conclusões

As abordagens que envolveram técnicas de probabilidade e distancia realmente mostram resultados melhores que as demais, como o esperado. Para o uso de tf-idf como forma de extração de dados de textos. Através de métodos que utilizam a distancia para classificar dados fica fácil distinguir as classes, dado a visibilidade que o algoritmo KNN tem sobre as distancias entre as frequências de termos em classes distintas. O tipo de pré-processamento deverá ser revisto em futuros trabalhos, uma vez que o modelo tf-idf não se adaptou muito bem a todos os classificadores. Não só o calculo do tf-idf deverá ser revisto mas também deve-se pensar na extração de mais características antes da transformação dos documentos em valores numéricos. Apesar do uso de Stop Words e remoção de numerais e alguns símbolos, muitos *tokens* dispensáveis para a classificação dos documentos continuaram presentes ao fim do pré-processamento. Muitos símbolos passam despercebidos pela procedimento de remoção e muitas palavras que não foram contabilizadas como StopWord poderiam ter sido ignoradas sem interferir na classificação, melhorando o desempenho dos classificadores e evitando que estas palavras, que podem se repetir em demasia em vários documentos, contabilizem como relevantes para a analise de sentimento. Talvez se um pré-processamento mais criterioso tivesse sido realizado, os resultados tivesse sido melhores e ao mesmo tempo, o corte excessivo de *features* poderia fazer com que os resultados ficassem piores.

Outro fator que pode ser testado em futuros experimentos é a remoção de elementos com frequências menores que um determinado valor 'n', da tabela de TF-IDF. A própria função do sklearn promove essa remoção ao adicionar um valor ao parâmetro 'n' e alteração desse parametro pode representar uma mudança significativa ou não.

Existem muitos fatores de pré-processamento que podem influenciar no resultado final de um algoritmo de analise de sentimento e todos eles podem e devem ser discutidos e analisados futuramente com mais perícia, quanto a abordagem TF-IDF, constata-se nesse estudo que a mesma possui melhores resultados com classificadores estatísticos/ pirobalísticos.

Referências

Aggarwal, C. C. and Zhai, C. (2012). *Mining text data*. Springer Science & Business Media.

Godbole, N., Srinivasaiah, M., and Skiena, S. (2007). Large-scale sentiment analysis for news and blogs. *Icwsm*, 7(21):219–222.

Hearst, M. (2003). What is text mining. *SIMS, UC Berkeley*.

Leung, C. W. (2009). Sentiment analysis of product reviews. In *Encyclopedia of Data Warehousing and Mining, Second Edition*, pages 1794–1799. IGI Global.

Muhr, T. (1991). Atlas/ti—a prototype for the support of text interpretation. *Qualitative sociology*, 14(4):349–371.

Pozzi, F. A., Fersini, E., Messina, E., and Liu, B. (2016). *Sentiment analysis in social networks*. Morgan Kaufmann.

Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.

SANTOS, F. C. et al. (2009). *Variações do método kNN e suas aplicações na classificação automática de textos*. PhD thesis, Dissertação de Mestrado, Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, Universidade Federal de Goiás, Goiânia, Brasil.