Críticas de filmes com mineração de texto Preparação dos dados e análise de sentimento.

Artur Henrique Brandão de Souza
15/0118783
Universidade de Brasília - UNB
Campus Darcy Ribeiro - Asa Norte
artur_henriquebs@hotmail.com

Resumo

Este artigo tem como objetivo explicar a metodologia utilizada para a captação da base de dados, o pré-processamento utilizado para o tratamento e transformação destes dados e classificar o texto através da análise de sentimento de palavras. Há, então, o intuito de aprofundar o conhecimento em estratégias para preparação dos dados envolvendo mineração de texto, para que, a partir deles, seja possível fazer análises mais assertivas quanto à técnica desejada ser aplicar ao final com o objetivo de classificar o texto. A base de dados escolhida foi uma coleção de *reviews* a respeito de filmes em prol de conseguir selecionar, ao final, qual o sentimento relatado pelo autor através das palavras utilizadas pelo mesmo durante a escrita de suas opiniões. Estes sentimentos são compostos por ter achado o filme ruim, bom ou regular. Será visto com mais detalhes nas sessões seguintes.

1. Introdução

Diversos sítios na internet dispõem da divulgação de análises de especialistas e opiniões próprias de autores que tiveram o intuito de criticar filmes, em que elas podem ter sido em um tom de reprovação para quando o filme, em sua maior parte, não tiver agradado o autor; de neutralidade para quando houver tanto lados positivos e negativos se equiparando no filme ; de aprovação quando, em sua maior parte, tiver agradado o autor.

Há diversos métodos utilizados pelos próprios autores para caracterizar um filme, sendo que os profissionais procuram criticar por meios mais técnicos, ou seja, observam a trilha sonora; enredo; efeitos especiais; entre outros e como todos esses métodos foram construídos e correlacionados no decorrer do filme. Enquanto, há também, autores que utilizam as mesmas características para criticar o filme, porém utilizando um viés mais pessoal, em que não é importante citar muitos detalhes técnicos acerca do objeto criticado.

Assim, a partir da coleta desses textos, neles serão trabalhados uma procura de palavras chaves que possam demonstrar um certo sentimento quanto ao foi escrito. Para isso, será necessário que haja um pré-processamento nos dados, ou seja, haverá transo formações nos dados, em que ocorre após a coleta dos dados e antes destes serem analisados, para que a análise final tenha um viés mais assertivo. Com isso, após feito o pré-processamento, os textos serão classificados quanto a que tipo de sentimento o autor transpareceu a partir das palavras utilizadas durante a escrita da crítica.

Em outras literaturas já foram feitas avaliações sobre o tema tratado. Porém, como os dados são baseados em textos, não há, portanto, uma garantia de cem por cento de eficácia quanto ao resultado final. Há diversos métodos de pré-processamento diferentes utilizados por diferentes autores, o que não torna o problema perto de estar resolvido.

O método proposto para a análise de sentimento das *reviews* dos filmes será tratado com a busca de palavras que expressam sentimento no decorrer dos textos. Assim, será avaliado a quantidade de ocorrência dessas palavras para que possa ser classificado como uma crítica negativa, positiva ou neutra quanto ao filme em questão.

Por fim, todo o processo que levará a chegar a tais conclusões contribuirá para futuras melhorias no pré-processamento dos dados e conseguir, por fim, estender a quantidade de sentimentos, sendo mais específico e detalhado quanto a forma que o filme foi criticada e conseguir verificar a opinião pública quanto ao filme em questão.

2. Revisão de Literatura

Em respeito ao texto *Sentiment Analysis and Classification Based On Textual Reviews* o problema a ser resolvido é a classificação dos *reviews* de filmes quanto a se foram criticas boas ou ruins. Para isso, foram utilizados métodos como *Bag of Words*(BOW) e *Support Vector Machine*(SVM) para o pré-processamento dos dados e através de um conjunto

de palavras chaves vinculadas para cada um dos sentimentos, foi feita uma classificação dos textos. Além disso, foi proposto o uso de um novo algoritmo chamado *Sentiment Fuzzy Classification* que utiliza métodos estatísticos para melhorar a acurácia do classificador.

Como resultado foi comparado que com a utilização do novo algoritmo, estatisticamente, tornaria o classificador obter uma acurácia maior. Um dos maiores obstáculos para isso é que a base de dados que se trata de textos, ou seja, todo a transformação do dado é complicada e, principalmente, criar uma base de dados para treinamento voltado para a analise de sentimento.

Já analisando o texto *Learning Word Vectors for Sentiment Analysis* este por si já procura trabalhar com a base de dados em si, focando no problema que se tem quanto a quais palavras realmente representam informações de sentimento. Para isso, foi-se criado um modelo probabilístico que é baseado na captura de semelhanças semântica das palavras. Como resultado foi comparado com outros modelos de algoritmos e este se saiu melhor na acurácia em sentenças que tinham como viés um lado mais subjetivo do autor. No entanto, uma de suas limitações era justamente o quando não se tratava de um texto com termos técnicos, mas sim quando este priorizava um lado mais subjetivo e mais pessoal do autor.

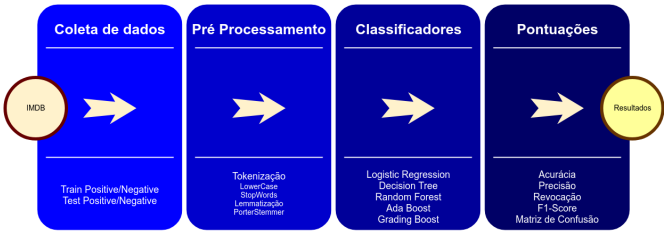
Além disso, com base no artigo *Movie Review Mining and Summarization* a ideia principal do mesmo era conseguir distinguir, através dos textos escritos nas críticas dos filmes, quais palavras eram destinadas para quais áreas da criação do filme, como para pessoas: diretor; atores; entre outros e para elemtentos como: trilha sonora; design dos personagens; efeitos especiais; entre outros. O método utilizado foi na criação de uma lista de palavras chaves que conseguiam distinguir para qual objeto uma determinada sentença do texto a pessoa estava se referindo.

Como resultado foi comparado, em 11 filmes, com o classificador utilizado chamado *Hu and Liu* em que se teve como diferencial um F1-score maior, em até 8.4 por cento, melhora em desempenho e uma precisão maior quando há mais de uma palavra junta a ser analisada. Contudo, esse método utilizado tem como obstáculo ambiguidade que há em diversas palavras, o que dificulta na classificação do texto.

3. Método Proposto

Nesta seção será abordada todo o processo, em detalhes, utilizado para a criação dos resultados finais. Assim, tem o objetivo de explicitar tanto da captação dos dados até a amostra de resultados feitos pelas acurácias dos classificadores, como visto na Figura1 através de um fluxograma.

Figura 1. Fluxograma classificação de textos para análise de sentimento



3.1. Coleta de dados

Para a coleta de dados, foi utilizado uma pasta, adquirida através de um site[3], em que nesta pasta há 50 mil arquivos textos em que cada uma contém uma opinião dita por alguma pessoa acerca de um filme. Esses arquivos estão divididos em quatro pastas, contendo 12,5 mil arquivos em cada. Primeiramente, foi divido em duas pastas em que separaram em 25 mil arquivos em cada que diz respeito a dados para que sejam utilizados para treino do classificador e dados para que possam ser utilizados para teste. Basicamente, dados de treino será utilizado para que o classificador consiga pegar esses dados e utilizar como modelo para conseguir classificar outro arquivo que vier aparecer. Já os arquivos para teste serão utilizados para verificar se dessa forma o classificador, através dos arquivos para treinos passados, consegue classificar corretamente um texto que por ventura vier aparecer. Por fim, cada pasta foi dividida em mais duas partes, uma contendo apenas os arquivos em que a nota do *review* foi acima de 5, o que quer dizer que comentaram bem a respeito do filme, e por outro lado com os arquivos que contém notas menores que 5 que condiz aos comentários negativos. Como consequência dessa separação, ficou bem mais fácil conseguir pré-definir os valores que serão utilizados para que o classificador consiga destinar um texto para seu respectivo resultado final, classificando-os como resultados positivos ou negativos conforme as palavras utilizadas pelo autor.Isso ocorre já que o dados rotulados, ou seja, sabemos das respostas finais desses dados, possibilitando assim que haja um treinamento, com dados reais, dos classificadores utilizados.

3.2. Pré Processamento

O pré-processamento feito nos textos serve para facilitar e melhorar a performance dos classificadores, em vista que os dados, que estão em formato de texto, estão vindo com todos os valores possíveis e isso pode ser que atrapalhe o classificador a rotular de forma correta os arquivos. Para isso é necessário que seja feito modificações nos textos para que estes sejam possíveis serem analisados e por fim melhorar o desempenho dos classificadores.

3.2.1 Tokenização

Cada arquivo contém um texto que é composto por um conjunto de sentencias que consequentemente é composto por um conjunto de palavras. Assim, para que seja mais fácil de se trabalhar com o texto, será necessário que o texto seja separado até chegar ao nível de palavras. Com isso, cada palavra passará a ser identificada como um *token*, ou seja, um bloco provindo do texto.

3.2.2 LowerCase e StopWords

Após a separação por palavras do texto, será utilizado uma técnica chamada *Lowercase* que consegue pegar a palavra, em formato utf-8, e pegar cada caractere e transformar em minusculo. Isso de certa forma ajuda bastante na questão de padronização, em vista que, para o computador, os caracteres 'A' e 'a' são entendidos como diferentes, porém para uma analise melhor, necessita-se que sejam vistos como iguais.

3.2.3 StopWords

Em um texto há diversas palavras que são utilizadas mais como forma de conectivos do que elas terem um sentido realmente importante para a analise de texto. Assim, foi utilizado um conjunto de dados já construídos pela plataforma *NLTK* que contém 179 palavras em que normalmente são utilizadas como conectivos e até mesmo pontuações, como ponto final, entre outros. No entanto, por ser um conjunto pré-definido, não há todas as palavras que necessariamente deveriam ser retiradas de uma análise, assim, foram acrescentadas mais 24 palavras, totalizando 203 palavras retiradas de analise durante o análise.

3.2.4 Lemmatização

A lematização de uma palavra é basicamente conseguir reduzir as palavras flexionadas para a palavra raíz, que seria atingir a palavra originalmente contida no dicionário. Assim, um exemplo acerca disso seria a palavra "comer", que pode ser flexionada para "comendo", "comi", entre outras. O que manteria de certa forma a palavra originadora seria "comer"e é isto que buscará ser feito durante esse processo.

3.3. Classificadores

A classificação de um texto significa conseguir predizer em qual categoria ele pertence. Como visto anteriormente, as categorias para este problema já foram pré-definidas sendo destinadas ou para a *label* de que o texto traz um teor positivo, ou para a *label* que traz um teor negativo para o texto.

3.3.1 Logistic Regression

Regressão logística é um algoritmo de classificação que visa retornar um valor de probabilidade ao fazer o uso de uma função sigmoide para destinar os textos a um conjunto de classes pré-definidas.

3.3.2 K-Nearest Neighbor

K-Nearest Neighbor é um algoritmo de classificação que visa classificar um dado através da similaridade de seus vizinhos mais perto. Esse classificador de certa forma é caracterizado como não parametrizado e lento, ou seja, a estrutura do modelo é determinada a partir do *dataset* enviado. Assim, foi designado para essa função a contagem de três vizinhos iguais mais próximos para definir qual a classe destinada para o dado descrito e foi passado como parâmetro para ser utilizado o algoritmo *kd tree* em vista que o algoritmo bruto que é utilizado por *default*, utilizava muito a memória dando *overflow* ao utilizar.

3.3.3 Decision Tree

Basicamente, esse algoritmo de classificação se resume a ter as *labels* como sendo suas folhas, ou seja, estão no final da árvore. E, durante toda a árvore até chegar em fim nas folhas, essas pontos representam um conjunto de características observadas pelos dados de treino que caracterizam a classe final da *label*.

3.3.4 Random Forest

O classificador *Random Forest* consegue criar um conjunto de árvore de decisões(descrito anteriormente) a partir de um subconjunto aleatório escolhido dos dados de treino passados. Assim, ele é baseado através da média adquirida em cada árvore para decidir a categoria final destinada.

3.3.5 Ada Boost

Esse algoritmo trabalha através da estimação dos dados em que a cada tentativa de classificar, é feito uma nova cópia do classificador utilizando o mesmo conjunto de dados, porém os pesos das instancias classificadas erradas são alteradas para que o algoritmo se concentre em casos mais difíceis.

3.3.6 Grandient Boosting

Gradient Boosting é um algoritmo de classificação que visa ajustar as n classes de uma árvore de regressão ao gradiente negativo da função. Contudo, como o teste é feito por apenas duas classes como desino, há apenas uma unica árvore de regressão a ser induzida nesse contexto.

3.4. Pontuações

As pontuações servem para que possa ser medido a assertividade quanto aos classificadores utilizados e colocar em uma escala o quão eficientes os algoritmos foram quanto ao seu propósito. Assim, foram utilizados cinco técnicas como: Acurácia; Precisão; Revocação; F1-Score; Matriz de confusão.

Foi utilizado a técnica de *cross validation* para todos os classificadores com a intenção de evitar que ocorra o chamado *overfitting* que condiz em não ajustar corretamente os dados de treinamento, assim, perdendo toda a ideia dos dados.

Além disso, foram se utilizadas métricas para conseguir medir determinadas pontuações acerca dos classificadores, que será melhor detalhada na seção seguinte em Resultados. Essas métricas são: Acurácia; Precisão; Recall-Score; F1-Score; Matriz de Confusao.

4. Resultados Experimentais

Para os resultados dos experimentos feitos foi utilizado, como dito anteriormente, a base de dados do IMDB com 50 mil *reviwes* de filmes em que estão separados igualmente em 4 pastas, sendo que metade voltado para separar arquivos para treino dos classificadores e metade para teste e cada um deles contém, também, separação dos dados entre *reviews* positivas e negativas.

Assim, para o experimento, foi se utilizado um computador com as configurações de *Processador Intel Core i5-5200U CPU @ 2.20GHz x 4*, com *11,7 GiB memoria RAM* e tipo de sistema de *64-bit*.

Classificador	Acurácia	Precisão	Revocação	F1-Score
Ada Boost	0.80036	0.8236	0.7677	0.7936
Decision Tree	0.7216	0.7215	0.7220	0.7217
Gradient Boosting	0.8084	0.7781	0.8629	0.8183
KNN	0.5956	0.5734	0.7475	0.6489
Logistic Regression	0.8773	0.8823	0.8709	0.8765
Random Forest	0.7572	0.8053	0.6783	0.7363

Figura 2. Tabela com valores das métricas

A técnica de acurácia e precisão significam, respectivamente, em questão das classes a serem preditas obtiveram exatamente os valores que eram esperados ao final da classificação e a capacidade do classificador de não rotular como positiva uma amostra de dados que deveria ser negativa, ou o contrário.

Além disso, há a revocação(*Recall-score*) que é basicamente a solução da fórmula

$$(\frac{Nvp}{Nvp + Nfn})$$

em que *Nvp* significa a quantidade de números verdadeiros positivos e *Nfn* para a quantidade de números com falsos negativos. Tendo como melhor valor sendo próximo de 1 e o pior sendo próximo de 0.

Alem do mais, há o F1-Score que é utilizado quando há uma necessidade de balanço entre a precisão e a revocação e quando há um grande número de verdadeiros negativos. A formula utilizada para essa métrica é:

$$F1Score = 2 * (\frac{Precisão * Revocação}{Precisão + Revocação})$$

Assim, como método de avaliação, quando mais próximo do valor 1, melhor o classificador e quanto mais próximo do 0, pior ele é.

Por fim, há a Matriz de Confusão que é basicamente uma tabela que mostra as frequências de classificação para cada classe do modelo.Assim, as frequências mostradas nessa tabela são: Verdadeiro positivo; Falso positivo; Falso verdadeiro; Falso Negativo. As frequências que são positivas são ditas que foram preditas de forma correta, já as negativas condiz com as previsões erradas do classificador.

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Positivo	9016	3484
	Negativo	3073	9025

Figura 3. Decision Tree

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Positivo	10450	2050
	Negativo	4021	8479

Figura 4. Random Forest

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Positivo	10413	2087
	Negativo	2904	9596

Figura 5. Ada Boost

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Positivo	5547	6953
	Negativo	3475	9344

Figura 6. K Nearest Neighbor

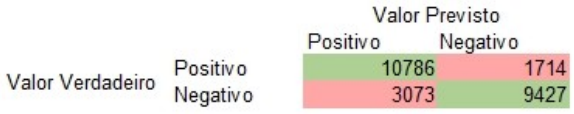


Figura 7. Gradient Boosting

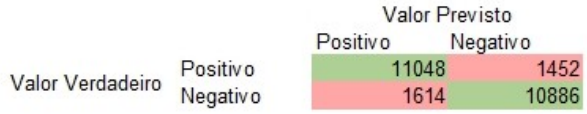


Figura 8. Logistic Regression

5. Conclusão

Como objetivos cumpridos foi visto em detalhes a questão da performance de modelos lineares de classificadores, verificando como seriam seus resultados a partir de um preparo do dado antes de colocar o mesmo para teste nestes classificadores, não fazendo com que recebessem dados brutos para o treinamento.

É notado, então, que um dos grandes problemas a serem trabalhados com os classificadores de modelos lineares, é principalmente com relação a baixa performance dos mesmos. Contudo, mesmo com toda a preparação e limpeza dos dados no período de pré-processamento, foi notória o baixo valor dos classificadores quanto a execução do *dataset* em questão.

Além disso, para futuros trabalhos, será de forma proveitosa que fosse estudado mais afundo quais as melhores técnicas para o pré-processamento, em vista que, como é trabalhado totalmente com texto, há toda uma questão informal de se esccrever o que pode trazer tons de ironia e ambiguidade que para certos tipos de tratamento são muito difíceis de serem trabalhados e classificados por classificadores lineares comuns, ou seja, sem contar com a parte de redes neurais.

Referências

[1] M. D. Ms.K.Mouthami and D. Bhaskaran. (2013) Sentiment analysis and classification based on textual reviews. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6508366

[2] F. J. Li Zhuang and X.-Y. Zhu. (2006) Movie review mining and summarization. [Online]. Available: https://dl.acm.org/citation.cfm?id=1183625

[3] P. T. P. D. H. A. Y. N. C. P. Andrew L. Maas, Raymond E. Daly. (2011) Learning word vectors for sentiment analysis. [Online]. Available: https://dl.acm.org/citation.cfm?id=2002491

[4] (2019) Python documentation. [Online]. Available: https://docs.python.org/3.5/

[5] (2019) Nltk documentation. [Online]. Available: https://www.nltk.org/

[6] J. W. Hatzivassiloglou . V. (2000) Effects of adjective orientation and gradability on sentence subjectivity, in: Proceedings of the international conferenceon computational linguistics (coling). [Online]. Available: pp.299\begingroup\let\relax\relax\endgroup[Pleaseinsert\PrerenderUnicode{}intopreamble]305

[7] (2019) Scikit learning documentation. [Online]. Available: https://scikit-learn.org/stable/documentation.html