Análise de Sentimento das Reviews do SFU Review Corpus

Mariana Silvestre e Weidmam Leles

Iscte-iul, Lisboa, Portugal
{mcseo,wmlss}@iscte-iul.pt

Resumo. Este relatório apresenta a aplicação de análise de sentimento ao *dataset* SFU_Review_Corpus que contém no total 400 *reviews*. A análise de sentimento foi realizada utilizando diferentes modelos de classificação: um modelo simples ao comparar com um léxico de sentimentos e modelos de aprendizagem automática. No modelo simples, criaram-se experiências onde se utilizou o tratamento da negação e algum pré-processamento. Enquanto que com a aprendizagem automática, testou-se vários cenários conjugando dois tipos de vectorização e várias formas de pré-processamentos, como a remoção da pontuação, *lowerization*, *tokenization*, *lemmatization* e *stemming*. Com isto, foi possível concluir que os modelos de aprendizagem automática com vectorização por contagem de palavras e com a combinação de diversos pré-processamentos apresentaram os melhores resultados.

Palavras-chave: Análise de Sentimento, Modelos de Classificação, Pré-processamento, Text Mining

1 Introdução

Perceber a opinião de uma pessoa pode parecer uma tarefa simples quando se trata de dois humanos a falar. No entanto, a análise de sentimentos realizada por um computador requer o tratamento computacional de opinião, sentimento e subjetividade num dado documento. Nesse sentido, ao combinar técnicas de Processamento de Linguagem Natural com técnicas de *Text Mining*, a perceção de opiniões e sentimentos requer a implementação de várias etapas complementares (Pange Lee, 2008 *apud* Martínez-Cámara, Martín-Valdivia, Molina-Gonzalez e Ureña-López ,2013).

É importante ter em conta que não basta criar classificadores ou usar classificadores já existentes, pois é necessário perceber todo o funcionamento dos modelos de classificação, assim como entender se será preciso realizar pré-processamento aos dados. No entanto, este pré-processamento poderá não dar origem a uma otimização do modelo, sendo que esta é vista através de métricas como a *accuracy*, *precision*, *recall* e *f-measure*. Sendo o SFU_Review_Corpus composto por um conjunto de dados balanceados, utiliza-se a *accuracy* como métrica de avaliação e comparação entre os modelos desenvolvidos.

Este relatório foi dividido em secções da seguinte forma: na secção 2 encontra-se uma breve revisão de literatura de forma a investigar o que já tinha sido realizado com o SFU_Review_Corpus; na secção 3 descreve-se os dados utilizados para a análise de sentimentos; na secção 4 pormenoriza-se todos os procedimentos que foram realizados aos dados para a análise; na secção 5 encontram-se os resultados obtidos nas diversas experiências e cenários realizados e por fim, na secção 6, exprimem-se as conclusões deste estudo e alguns aspetos que podem ser melhorados num próximo projeto.

2 Análise do Trabalho Relacionado

Indhuja e Reghu (2014) realizaram a análise de sentimentos utilizando a lógica *fuzzy*, passando pelas fases de pré-processamento, extração de *features* e classificação. Ao longo de toda a análise utilizaram técnicas NLP (*Natural Language Processing*), tal como *Named Entity Recognition*, *POS tagging* e análise sintática. Para além disso, os autores recorreram ao cálculo do *tf-idf* para identificar as palavras de maior importância. Com esta análise e através de indicadores como *recall*, *precision* e *f-measure* concluíram que o sistema teve um bom desempenho com uma precisão de aproximadamente 85%, sendo a lógica *fuzzy* um bom modelo para a análise de sentimento.

No caso do estudo desenvolvido por Natalia, Sheila, Noa, Manuel, Maite e Ruslan (2012) para além de técnicas como a tokenização e análise sintática, também utilizaram o tratamento da negação. Com o uso deste tratamento, verificaram algumas melhorias na classificação. À semelhança de Indhuja e Reghu (2014), utilizaram métricas como *f-measure* e *kappa* de forma a verificar se a concordância entre os anotadores era elevada. Através deste estudo, os

autores concluíram que "as diretrizes são sólidas e que o corpus será útil para a análise de sentimento e reconhecimento de negação".

Martínez-Cámara, Martín-Valdivia, Molina-Gonzalez e Ureña-López (2013) em *Bilingual Experiments on an Opinion Comparable Corpus* enfatizam que os sistemas baseados numa abordagem supervisionada são os mais bem sucedidos para a extração de opinião, pelo que se torna importante a aplicação de algoritmos de aprendizagem automática aquando da construção destes modelos de classificação. Além disso, destacam a importância da aplicação de pré-processamentos, como por exemplo, o *stemming*, o qual demonstrou ser capaz de melhorar a classificação nos comentários. Os autores também frisam que por se tratar de uma tarefa de extração de opinião, o tratamento da negação é uma atividade imprescindível.

Em suma, verifica-se que os autores dos diferentes artigos estão em concordância, embora sejam utilizadas diferentes técnicas, todos aferem que este corpus é útil e pode ser utilizado para a análise de sentimento.

3 Descrição dos Dados

O *SFU_Review_Corpus* é um *dataset* com dados balanceados, este conjunto é composto por duas colunas: a coluna *text* e a coluna *recommended*. Na coluna *text* encontram-se as *reviews*, as quais são classificadas como "*yes*" ou "*no*", na coluna *recommended*.

Este conjunto contém 400 *reviews* que estão divididas nas seguintes categorias: *Books*, *Cars*, *Computers*, *Cookware*, *Hotels*, *Movies*, *Music*, *Phones*. Por sua vez, cada categoria é composta por 25 *reviews* classificadas como *recommended* e 25 *reviews* classificadas como *not recommended*. Estas *reviews* já se encontram divididas em dois conjuntos: num conjunto de treino que contém 320 *reviews* e num conjunto de teste que contém as restantes 80 *reviews*.

As *reviews* foram obtidas do site *Epinions* em 2004, pelo que foram escritas por pessoas diferentes e os seus tamanhos são também diferentes.

4 Descrição das Tarefas e Procedimentos

4.1 Preparação dos dados e criação de uma baseline

Inicialmente, começou-se por fazer uma breve preparação dos dados. Para tal, realizou-se uma análise de sentimentos através de um classificador já existente, para isso utilizou-se a função *sentiment.polarity* da biblioteca *TextBlob*. A *accuracy* foi de 0,6375 pelo que se considera um risco moderado, ou seja, existem dados resultantes que poderão não corresponder à realidade.

4.2 Aplicação de um léxico de sentimentos

Nesta etapa, o objetivo principal era criar um classificador através da utilização de um léxico de sentimentos. Para isso, realizou-se duas experiências principais: Experiência 1 em que se utilizou os dados na sua forma original e uma Experiência 2 onde se aplicou o tratamento da negação, ou seja, foi adicionado às palavras o "NOT_" desde a palavra a seguir a uma negativa até se encontrar um sinal de pontuação.

Posto isto, realizou-se mais duas experiências onde se aplicaram duas técnicas de pré-processamento com a finalidade de melhorar os resultados obtidos. Na Experiência 3 mantiveram-se os dados originais onde se aplicou a *lemmatization* e o *stemming* e na Experiência 4, após se aplicar o tratamento da negação, aplicou-se também a *lemmatization* e o *stemming*.

Assim, para se classificar as *reviews* com um classificador simples, procedeu-se à separação das *reviews* por palavras para que se conseguisse verificar se cada palavra constava no léxico e qual a sua classificação. De seguida, somou-se as classificações por *review* e atribuiu-se a nova classificação, "*yes*" se a classificação total da *review* fosse igual ou superior a zero e "*no*" se fosse inferior. Nas experiências com tratamento da negação, utilizou-se o inverso, ou seja, se a palavra tivesse o "NOT_" e a palavra original (sem o "NOT_") estivesse no léxico a classificação era o valor inverso.

De forma a verificar qual a melhor experiência calculou-se a *accuracy* de cada uma, onde se comparou as classificações reais com as classificações previstas. Os resultados serão demonstrados mais à frente na secção dos resultados obtidos deste trabalho.

4.3 Aprendizagem Automática

Com o objetivo de melhorar os resultados dos modelos de classificação desenvolvidos na etapa anterior, recorreu-se à metodologia de aprendizagem automática das ferramentas *scikit-learn* (SKLEARN) e *Natural Language Toolkit* (NLTK). Deste modo, percorreu-se as seguintes etapas: (I) construção de modelos inicias; (II) aplicação do classificador ao conjunto de teste; (III) avaliação e ajuste dos parâmetros; (IV) aplicação de pré-processamentos e construção de cenários.

Construção de modelos iniciais e aplicação do classificador ao conjunto de teste (I e II)

Durante a primeira etapa da aprendizagem automática, foram desenvolvidos dois modelos por meio da biblioteca NLTK: *Naive Bayes* (NBC) e *Maximum Entropy* (MEC). Para a construção destes modelos realizou-se a tokenização, já que este processo é importante para ajudar a perceber o significado dos comentários ao analisar as palavras presentes no texto. De seguida, selecionou-se as 1500 palavras mais importantes por meio do *tf-idf* (*term frequency–inverse document frequency*). Por fim, o processo de aprendizagem automática contou com 15 interações.

No que se refere a biblioteca *scikit-learn*, inicialmente foram desenvolvidos dez modelos: cinco por meio da vectorização baseada no *tf-idf* [3], qual considera o peso global de uma palavra num documento, e cinco por meio da vectorização baseada em contagem [4], que conta o número de vezes que uma palavra aparece num conjunto de dados. Deste modo, implementou-se algoritmos de Regressão logística, *Multinomial Naive Bayes*, *Gaussian Naive Bayes* e *Linear Support Vector Classification*. Ainda assim, é importante notar que para todos os modelos de classificação utilizando a biblioteca *scikit-learn* extraiu-se 1500 *features*, aplicou-se o *document frequency* mínimo de 3 e máximo de 70% e utilizou-se *stop-words*, que são palavras presumivelmente não informativas. A seleção destes parâmetros foi feita através de tentativa-erro, que se encontra descrita de forma mais pormenorizada na etapa seguinte.

Avaliação e ajuste dos parâmetros (III)

Uma vez construídos os modelos iniciais, perceber o quão bom são estes modelos é um processo determinante para o sucesso da aprendizagem automática. Portanto, o processo de avaliação dos modelos é uma tarefa importante tanto para a definição de quão boas são as previsões como para indicar a necessidade de melhorar o modelo ou até mesmo de construir novos modelos.

Conforme mencionado anteriormente, a avaliação ocorreu pela comparação da *accuracy* dos modelos iniciais, prevalecendo os modelos com os valores mais altos. No entanto, torna-se oportuno destacar que ao construir um modelo de aprendizagem automática é necessário ter em mente que diferentes problemas exigem diferentes abordagens e ferramentas, pelo que a seleção do melhor modelo somente com base na *accuracy*, que foi o método escolhido para este trabalho, pode ser uma abordagem limitada em algumas situações.

Ao comparar as *accuracies* dos modelos iniciais das ferramentas *NLTK* e *scikit-learn*, decidiu-se dar continuidade com a segunda ferramenta no sentido de refinar a sua utilização, proceder à validação cruzada aprimorando os parâmetros e realizando pré-processamentos para a criação dos dez cenários referidos anteriormente. Além disso, a ferramenta NLTK não oferece suporte direto à validação cruzada para algoritmos de aprendizagem automática, fator também decisivo para a escolha da biblioteca a ser utilizada nos passos seguintes.

Tal como o processo de avaliação, a parametrização é outra tarefa de grande importância para que se consiga elaborar bons modelos de classificação. Neste trabalho, o processo de escolha dos parâmetros foi feito através da metodologia de tentativa e erro. Isto é, definir inicialmente um valor arbitrário para um parâmetro, observar como o modelo se comporta e voltar a tentar outro valor para este parâmetro e assim sucessivamente até que se encontre os valores mais adequados.

Mediante as tentativas realizadas, os parâmetros para a comparação de todos os modelos de classificação utilizando a biblioteca *scikit-learn* são os seguintes: 1500 *features*, *document frequency* mínimo de 3 e máximo de 70%, utilizou-se também a lista *built-in* de s*top-words.*

Aplicação de pré-processamentos e construção de cenários (IV)

Nesta etapa procurou-se processar os dados não estruturados de forma a torná-los mais legíveis para que os algoritmos conseguissem interpretá-los de maneira mais eficaz. Ou seja, a estratégia utilizada para a escolha do pré-processamento a aplicar foi no sentido de se identificar o significado dos termos nos comentários, reduzir a

dimensionalidade dos nossos dados e eliminar ruídos. Deste modo, apostou-se na combinação das seguintes técnicas: Remoção da pontuação, *Lowerization*, *Tokenization*, *Lemmatization* e *Stemming.*

No entanto, sabe-se que a aplicação destes pré-processamentos não é uma garantia de que os modelos serão melhores, pelo que se torna pertinente a construção de cenários com combinação de diferentes métodos. Assim, a cada um dos algoritmos, o primeiro deles com a vectorização baseada no *tf-idf e* o segundo com a vectorização baseada em contagem de palavras, para além de um cenário sem nenhum pré-processamento, foram aplicados os mesmos quatro cenários:

Cenário 1 – aplicou-se a remoção da pontuação, *lowerization*, *tokenization*, *lemmatization* e *stemming*;
Cenário 2 – aplicou-se a remoção da pontuação, *lowerization*, *tokenization* e *lemmatization*;
Cenário 3 – aplicou-se a remoção da pontuação, *lowerization*, *tokenization* e *stemming*;
Cenário 4 – aplicou-se a *tokenization* e o *stemming.*

Além disso, é importante ressalvar que para cada cenário foram construídos modelos de classificação por meio da Regressão logística, *Multinomial Naive Bayes*, *Gaussian Naive Bayes* e *Linear Support Vector Classification*.

5 Resultados Obtidos

5.1 Aplicação de um léxico de sentimentos

Ao olhar para os resultados obtidos, foi possível verificar que quando se aplica o pré-processamento a classificação prevista é melhor tanto nas experiências sem negação como nas experiências com negação. Contudo, é visível que o salto maior ocorreu nas experiências sem o tratamento da negação, isto pode ocorrer visto que quando se aplica o tratamento da negação podemos ficar com frases inteiras com classificação negativa o que acaba por equilibrar o número de palavras positivas e negativas numa *review.*

Com a aplicação do pré-processamento é normal os valores da *accuracy* terem subido, visto que a *lemmatization* e o *stemming* acabam por uniformizar as palavras, ou seja, acabam por levar, em alguns casos, a uma diminuição o número de *features*. Assim, nota-se que fica mais fácil de encontrar as palavras no léxico.

Tabela 1. Resultados obtidos nas experiências com aplicação de um léxico de sentimentos.

Experiências	Tratamento Utilizado	Accuracy
Experiência 1	Sem Negação	0,525
Experiência 2	Com Negação	0,600
Experiência 3	Sem Negação + Lemmatization + Stemming	0,625
Experiência 4	Com Negação + Lemmatization + Stemming	0,613

5.2 Aprendizagem automática

No que se refere aos resultados obtidos por meio da aprendizagem automática, nota-se que o melhor modelo foi o Cenário 1 da vectorização baseada em contagem de palavras, ou seja, o modelo em que foi aplicado a maior quantidade de pré-processamentos. No entanto, os resultados assinalam também que o emprego de pré-processamentos por si só não é uma garantia de melhoria na *accuracy*, como ocorreu no Cenário 1 da vectorização baseado no *tf-idf*, que teve um resultado mais baixo do que os modelos sem pré-processamentos.

Assim, é possível perceber que encontrar a representação numérica do *SFU_Review*_Corpus por meio da vectorização baseada em contagem de palavras é a forma mais eficiente, já que todas as experiências que utilizaram esta forma tiveram um resultado mais promissor, conforme é possível observar na Tabela 2.

Tabela 2. Média das Accuracies das Validações Cruzadas (k = 10) para cada cenário

Cenários Realizados	Accuracy Média Das Validações cruzadas (k =10)
NLTK – NBC e MEC	0,568 – Sem validação cruzada
SKLEARN – CountVectorizer – sem pré-processamento	0,734
Cenário 1	0,756
Cenário 2	0,745
Cenário 3	0,748
Cenário 4	0,734
SKLEARN – TfidfVectorizer – sem pré-processamento	0,736
Cenário 1	0,727
Cenário 2	0,728
Cenário 3	0,726
Cenário 4	0,702

Para realizar esta comparação calculou-se os *scores* através da validação cruzada (K=10), para todos modelos em cada um dos cenários. Posteriormente, utilizou-se a média de cada validação cruzada para se calcular a média da accuracy de cada cenário.

Com o intuito de promover uma comparação mais visual e intuitiva, gerou-se o gráfico da Figura 1, através do qual podemos aferir que a combinação de elementos é fundamental, porém não a garantia de um melhor modelo.

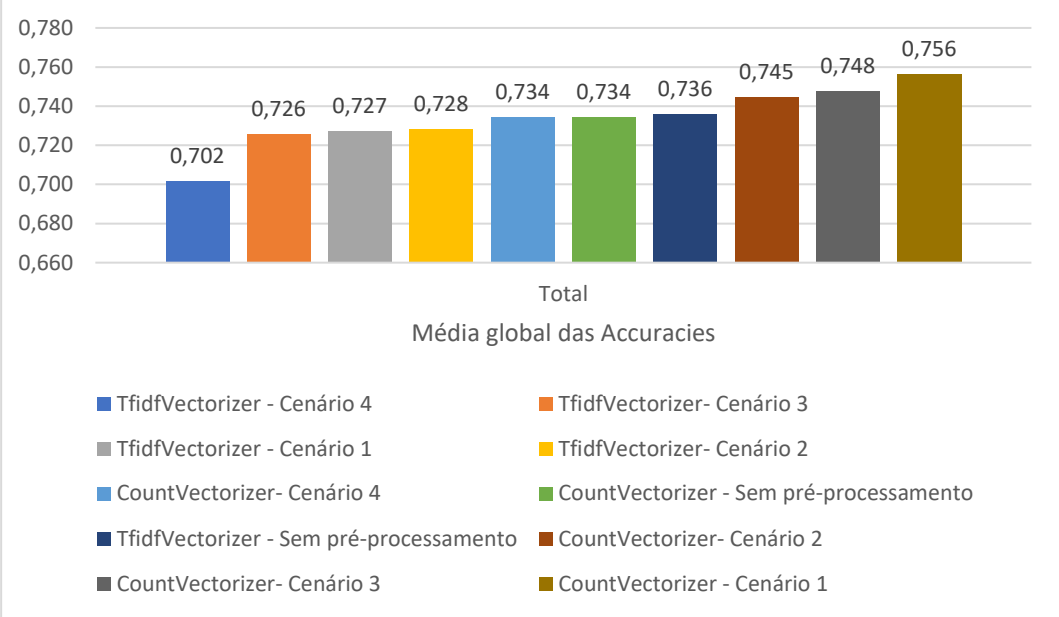

Fig. 1. Comparação das Médias Globais das Accuracies das Validações Cruzadas para cada experiência

6 Conclusões e Aspetos a Melhorar

Com base nas reviews do *SFU_Review_Corpus*, no decorrer deste trabalho foram desenvolvidos diversos modelos de classificação de forma a ser possível categorizar se uma *review* está a recomendar ou não um produto ou serviço.

Para se atingir este objetivo, investiu-se em três metodologias distintas para obter um bom classificador, o qual foi avaliado com base na sua *accuracy*. Assim, usufruiu-se de um classificador já existente, o *TextBlob*, produziu-se

quatro experiências através da aplicação de um léxico de sentimentos e por fim, elaborou-se doze modelos por meio da aprendizagem automática.

A elaboração e implementação de cada uma destas abordagens levou em consideração procedimentos particulares com o intuito de se obter o maior índice de acertos possível, tendo em vista a especificidade de cada metodologia. Não obstante das forças e fraquezas de cada um dos modelos desenvolvidos, ao compararmos o melhor modelo de cada uma dessas três abordagens, o modelo de aprendizagem automática produz o melhor resultado de todos: 21% maior do que o modelo com a aplicação de um léxico de sentimentos e 18,5% mais assertivo do que o modelo do *TextBlob*.

Em síntese, constata-se que a análise de sentimento é, para além de toda ciência por trás do *text mining*,um exercício experimental. À vista disso, mesmo que existam algoritmos mais adequados para uma determinada situação, efetuar diferentes experiências pode gerar bons resultados, uma vez que tendo em conta a diversidade das caraterísticas de cada conjunto de dados, distintas metodologias produzem diferentes respostas para os vários *inputs.*

Neste sentido, um dos aspetos a serem melhorados é a utilização de múltiplas combinações de pré-processamentos e a aplicação de outras formas de aprendizagem de máquina, como por exemplo, as redes neuronais artificiais.

Este trabalho foi realizado por dois elementos, pelo que consideramos que a divisão, em percentagem, foi a seguinte: Mariana Silvestre (50%) e Weidmam Leles (50%).

7 Bibliografia

1. Martínez-Cámara, Eugenio & Martín-Valdivia, Maria & González, M. Dolores & López, L. Bilingual Experiments on an Opinion Comparable Corpus. (2013).
2. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12 (2011).
3. Sklearn.Feature_Extraction.Text.Tfidfvectorizer — Scikit-Learn 0.24.1 Documentation". Scikit-Learn.Org. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html, last accessed 2021/04/17.
4. "Sklearn.Feature_Extraction.Text.Countvectorizer — Scikit-Learn 0.24.1 Documentation". Scikit-Learn.Org. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorize r.html, last accessed 2021/04/17.
5. Cruz Diaz, Noa & Konstantinova, Natalia & Castilho, Sheila & Maña, Manuel & Taboada, Maite & Mitkov, Ruslan. A review corpus annotated for negation, speculation and their scope. http://www.lrec-conf.org/proceedings/lrec2012/pdf/533_Paper.pdf. (2012).
6. I K. Indhuja and R. P. C. Reghu, "Fuzzy logic based sentiment analysis of product review documents," 2014 First International Conference on Computational Systems and Communications (ICCSC), Trivandrum, India.https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7032613. (2014).