Sentiment Analysis of Mobile Reviews

Abhinav Nigam, Karthik SK, Sasikiran Karri, Satyam Kumar, Siva Kranthi Kumar Mallipeddi,
Sreedhar Reddy Vundela, Sudhakar Kulkarni Mukayya, Sumit Sinha

DA-203 ICAIML

IISC, Bangalore

***Abstract*— In recent years, the rapid increase in smartphone and tablet usage has meant that the customers are able to provide reviews for every product or service that they use in a very easy and convenient way. This has led to a proliferation of reviews of all kinds of products and services, thus providing organisations with a very effective tool of measuring the performance of their products, in real time, without investing in sampling and user surveys.**

However, to take full advantage of this new treasure trove of data, organisations need good tools that interpret large quantities of data, and convert them into actionable insights for their product, marketing, logistics and other functions. In this paper, we have used a number of NLP analysis and inference techniques to get user sentiment insights from the mobile reviews dataset that we have collected from Amazon and Flipkart.

***Keywords*— Sentiment, Polarity, Mobile, Phone, NLP**

I. Methodology/DATA

The mobile user review data has been collected from two sources.

1. Scrapping the Flipkart website. This was built using scrapy, the code for this is present here: https://github.com/DSBA-IISc-Team8/flipkart-mobile-reviews-scraping
2. Amazon mobile review dataset from Kaggle. This is available at the following link: https://www.kaggle.com/datasets/nehasontakke/amazon-unlocked-mobilecsv

Fig. 1 Word Cloud

A. Clean-up

The following was done as part of data cleanup:

- The datasets were converted from JSON format to CSV format.
- The irrelevant columns were dropped. (Price, title, likes, dislikes) was dropped from the Flipkart dataset and (Brand name, review votes, price) was removed from Amazon dataset.
- The remaining column names were then normalised between the two datasets.
- The two datasets were then merged.

B. *Pre-processing of data*

The following pre-processing steps were performed on the dataset:

1. Punctuations, HTML links, spaces, contractions have been removed from the dataset, using a regular expression for special characters
2. Stop words have been removed from the dataset, using the stopword corpus from nltk.
3. All characters in the dataset have been converted to lowercase.

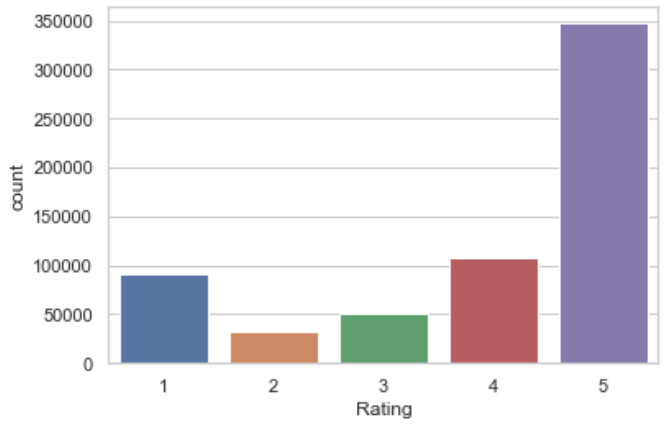

Fig. 2 Ratings distribution

II. Experimentation

The following techniques were experimented with, with the aim of getting the best f1 score.

A. TF-IDF

TFIDF or *term frequency - inverse document frequency* is a way of measuring the importance of a word within a corpus of documents.The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word. We used the TfIdfVectoriser to create the tf idf dataset from our corpus, and then trained a LinearSVC model with our corpus. Further we explored using Logistic Regression for improvement.

B. N-Gram And LSTM

N-grams are a contiguous stream of N words extracted from a sentence. For example, given a sentence:

I love machine learning

The 2-grams for the sentence will be the set S consisting of following tuples:

(*I, love*), (*love, machine*), (*machine, learning*)

This method is an improvement over the bag of words method, as it captures more meaning that is only revealed when two or more words are used together.

We vectorised our dataset into 1-gram, 2-gram and 3-gram and then trained a Multinomial NaiveBayes classifier using the vectors created. GridSearchCV was used to discover the best combination of hyperparameters, *alpha and fit_prior.*

LSTMs are explicitly designed to avoid the long-term dependency problem. Our model consists of an embedding layer, LSTM layer and a Dense layer which is a fully connected neural network with sigmoid as the activation function. Dropouts are added in-between layers and also on the LSTM layer to avoid overfitting. Use word embeddings. for capturing the context of a word in a sentence or document

C. BERT

BERT or Bidirectional Encoder Representations from Transformers is a transformers based technique for pretrained NLP. The bidirectionality in BERT implies that it can read data in both directions..The framework was pre trained using Wikipedia dataset, and since then there are a number of versions of BERT trained in various contexts.

The version of BERT we used is the bert-base-multilingual-uncased-sentiment. This has been especially trained on product reviews in six languages including English, and is widely used for sentiment analysis, with 1.2 million downloads in the last 30 days.

D. Feature recognition and sentiment analysis

We were seeing reviews in which people were talking about multiple aspects of a phone, for example the following review:

"amazing smartphone camera battery back display awesome little bit heating problem amazing smartphone students becoz im using student heavy also bad point smartphone good and i'm satisfied delivery also good thanks flipkart"

This review talks about camera, battery backup and display positively, but highlights a problem

with the phone heating up. To capture the sentiment for individual aspects of the phone, we tried out aspect/topic based sentiment analysis.

To achieve this, we first split the review into multiple sentences, and then ran it through Spacy's nlp() method, which provides the type (Noun, Verb, Adjective) for each token in the dataset, along with the dependency for each token. We then use a heuristic algorithm to determine the aspects of a sentence, and the adjective associated with that aspect. The adjectives are then passed to a TextBlob class, to get the polarity and subjectivity score for an aspect, which is then converted into a sentiment score.

III. *Results*

Technique	**Metrics**					
	Accuracy	**Precision**	**Recall**	**F1-score**	**Hyper parameters**	**#features**
N-grams(1)	0.6	0.52	0.61	0.53	{'alpha': 1.0, 'fit_prior': True}	88280
N-grams(2)	0.6	0.51	0.69	0.51	{'alpha': 0.75, 'fit_prior': True}	1544246
N-grams(3)	0.6	0.51	0.60	0.49	{'alpha': 0.75, 'fit_prior': True}	4820742
TF-IDF(Logistic)	0.53	0.43	0.53	0.41	{'logistic__C': 10}	127874
LSTM	0.8*	0.96	0.8	0.87	dropout:0.5, activation: sigmoid	
BERT	0.96 * 0.53 **				Epochs:3, bacth_size: 4	

*: At sentiment level (_ve, -ve & neutral)
**: At rating level (1, 2, 3, 4 & 5)

Github Code:
https://github.com/DSBA-IISc-Team8/sentiment-analysis-of-mobile-reviews

References

[1] Aspect based sentiment analysis - Aris Pattakos
[2] Feature engineering for NLP in Python - datacamp.com
[3] bert-base-multilingual-uncased-sentiment - huggingface.co