Opinion Mining

Atasie Oana-Andreea, gr. 341

Ilieș Elena, gr. 341

Petre-Vlad Mihaela, gr. 241

Contents

Introduction

The chosen topic is Opinion mining - the implementation of an algorithm that classifies the reviews of a product after the general opinion/rating.

We used the following dataset:

https://data.world/opensnippets/amazon-mobile-phones-reviews

The dataset is a list of over 30,000 reviews for brands like Apple, Samsung, OnePlus and Redmi. The chosen product for review extraction is OnePlus Nord 5G (Gray Onyx, 8GB RAM, 128GB Storage).

Analysis of main idea

Opinion mining is a text analysis method that uses computational linguistics and natural language processing to identify and extract people's opinions and sentiments within a text (positive/good, negative/bad, neutral, etc.). In terms of applicability, opinion mining is widely used in domains like social media or customer reviews, to determine and categorize opinions about a product, service, or idea.

The most common use of opinion mining is to categorize comments and statements on a simple scale of opinion polarity: positive, neutral and negative.

Challenges with opinion mining

- Objectivity or comments with a neutral sentiment tend to pose a problem for systems and are often misidentified.
- Computer programs have trouble when encountering emojis and irrelevant information. Special attention needs to be given to training models with emojis and neutral data so as to not improperly flag texts.
- People can be contradictory in their statements. Most reviews will have both positive and

negative comments. The more informal the medium, the more likely people are to combine different opinions in the same sentence and the more difficult it will be for a computer to parse.

- Irony and sarcasm often cannot be explicitly trained and lead to falsely labeled sentiments.

Some of the most popular types of opinion mining

- **Fine-grained** opinion mining provides a more precise level of polarity by breaking it down into further categories, usually very positive to very negative. This can be considered the opinion equivalent of ratings on a 5-star scale and is commonly used in opinion polls or surveys.
- **Emotion detection** identifies specific emotions rather than positivity and negativity. Examples could include happiness, frustration, shock, anger and sadness.
- **Aspect-based analysis** gathers the specific component being positively or negatively mentioned.

Related work

In "Opinion Mining of Online Customer Reviews"[1] written by Patlammagari Gowtamreddy, it is said that opinion mining has become a fascinating research area due to the availability of a huge volume of user-generated content in review sites, forums and blogs. Opinion mining has applications in a variety of fields ranging from market research to decision making to advertising. In this thesis it was shown how Apriori frequent item set mining algorithm can be used in Opinion Mining process and how normalized polarity varies for the values taken from customer reviews star ratings and from values of opinion mining process that uses Apriori algorithm and SentiWordNet. Apriori algorithm refers to an algorithm that is used in mining frequent products sets and relevant association rules. Generally, the apriori algorithm operates on a database containing a huge number of transactions. SentiWordNet is built via a semi supervised method and could be a valuable resource for performing opinion mining tasks: it provides a readily available database of term sentiment information for the English language, and could be used as a replacement to the process of manually deriving ad-hoc opinion lexicons.

In “Opinion mining for national security: techniques, domain applications, challenges and research opportunities”[2] written by Noor Afiza Mat Razali, Nur Atiqah Malizan, Nor Asiakin Hasbullah, Muslihah Wook, Norulzahrah Mohd Zainuddin, Khairul Khalil Ishak, Suzaimah Ramli & Sazali Sukardi, a structured literature review has been done on 122 articles in order to examine all relevant research accomplished in the field of opinion mining and the suggested Kansei approach to solve the challenges that occur in this domain. The Kansei approach can be defined as a product development methodology that translates customers' and users' feelings, impressions, and emotions into the domain of product design. The study in this article analyses various techniques of opinion mining as well as the Kansei approach that helps to enhance the techniques in mining people's sentiment and emotion based on text in cyberspace. Most of the study addressed methods such as machine learning, lexicon-based approach, hybrid approach, and the Kansei approach in opinion mining, which reported precise results for the assessment of human emotion. Therefore, this research suggests that the Kansei approach should be a complementary factor in the development of a dictionary focusing on emotion in the national security domain.

Description of experiments and methods used

Our team’s goal is to build a Machine Learning model that can identify the opinions expressed in a set of reviews with as high accuracy as possible.

Information about the dataset used to train and test our models

1. The whole dataset is a collection of reviews from Amazon for three phone models. We chose to use only the reviews for OnePlus Nord 5G (Gray Onyx, 8GB RAM, 128GB Storage). There are 9469 reviews in the dataset for this phone.

2. The dataset has a star rating system. Each review from the dataset has a whole number of stars, from 1 to 5. We annotated the opinions expressed in the reviews based on their number of stars. In our experiments we used two different annotations:

- A simple scale of opinion polarity with three categories: bad (1 and 2 stars), neutral (3 stars) and good (4 and 5 stars).
- Fine-grained opinion mining with five categories: from 'very bad' (1 star) to 'very good' (5 stars).

3. It is important to note that the dataset we used has a few flaws:

- There are duplicate reviews that have been eliminated. We used only the unique reviews - 9067 in total.
- The dataset is not balanced. Over half of the reviews (4558) have five stars and about 21% (1959) have 4 stars. That means approximately 71% of the reviews express a good opinion. The fewest reviews are the ones with 2 stars, only about 5% (456).

Methodology

Data preprocessing is the first step in building a Machine Learning model**.** This is an essential step that can greatly affect the performance of the model. However, preprocessing a text can be done using various methods and the only way to determine a good method is by trial and error.

The second step in building a model is choosing the method for classification. Again, there are many methods that one can use to classify data, but you cannot know for sure which is the best one.

Therefore, a good solution to a Machine Learning problem is found through experimentation. For this reason, our team tried different methods for data preprocessing and classification.

In the descriptions of our experiments the following notions will be used:

- **Lemmatization** – this is the process where we try to reduce the tokens from a sentence in a given language to their base form (the lemma), using a vocabulary and performing morphological analysis to remove inflectional endings. The lemmas are words that exist in the language.

- **Stemming –** it is a simpler version of lemmatization where we try to strip the suffix at the end of a token. It is faster than **Lemmatization**, but the output is not always a word that exists in the language.
- **Stopwords** - a set of commonly used words in a language that are commonly filtered out before or after processing of natural language data.
- **spaCy -** a free open-source library for Natural Language Processing in Python
- **Scikit-learn(sklearn)** - a free software machine learning library for the Python programming language
- **A multilayer perceptron (MLP)** is a fully connected class of feedforward artificial neural network (ANN).
- **Support vector machines (SVMs)** are a set of supervised learning methods used for classification, regression and outliers detection.
- **Tf–idf**, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. 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, which helps to adjust for the fact that some words appear more frequently in general.

Experiments

Note: In all the methods of preprocessing used all the words were turned to lowercase and the punctuation was eliminated. These steps won't be mentioned in the experiment descriptions.

1. The first experiment

The preprocessing was done using the following steps:

- Eliminated all links, hashtags, mentions and numbers from all reviews;
- Contracted words with apostrophes (for example don't was turned into don't)
- The tokens were extracted using nlp from spaCy;
- Eliminated the list of stopwords provided by spaCy, with the exception of the ones that can be used to denote a negative opinions: 'aren', "aren't", 't', 'but', 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'don', "don't", 'do', 'not', 'hadn', "hadn't", 'hasn', "hasn't",

'haven', "haven't", 'isn', "isn't", 'no', 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't", 'be', 'will', 'have'

- Applied lemmatization to the extracted tokens.
- Built a vocabulary of unigrams, bigrams and trigrams using CountVectorizer from sklearn, with the top 800 features ordered by term frequency. This is a bag-of-words model.
- The features were scaled using L2 normalization.

The classification was made using the **MLP and SVM with RBF kernel** from sklearn: 80% of the data was used for training and the rest was used for testing.

The MLP was used with the following parameters:

hidden_layer_sizes = (10), activation='relu', solver='adam', alpha=0.01, batch_size='auto', learning_rate='constant', learning_rate_init=0.01, n_iter_no_change=10, max_iter=300, tol=0.00001, shuffle=True, random_state=10, warm_start=True

The SVM was used with the following parameters:

C=1, kernel='rbf', gamma='scale', tol=0.0001, class_weight=None, max_iter=-1, random_state=10

The two classifiers were tested on the testing data divided into 5 categories, then into 3 categories an gave the following results:

- 5 categories:
  - MLP – Accuracy=54.52%, F1(weighted)=54.50%
  - SVM – Accuracy=0.62.51%, F1(weighted)=57.96%
- 3 categories:
  - MLP – Accuracy= 80.2%, F1(weighted)= 80.5%
  - SVM – Accuracy= 83.57%, F1(weighted)= 80.72%

In both cases the SVM performs better.

2. <u>The second experiment</u>

This experiment is similar to the first one, the only thing changed being the vocabulary built with CountVectorizer. This time the vocabulary consists of the top 600 bigrams and trigrams.

The results were worse than the first experiment:

- 5 categories:
  - MLP – Accuracy= 56%, F1(weighted)= 53.67%
  - SVM – Accuracy= 59.54%, F1(weighted)= 54.38%
- 3 categories:
  - MLP – Accuracy= 76.24%, F1(weighted)= 75.37%
  - SVM – Accuracy= 80.04%, F1(weighted)= 76.40%

Again the SVM performs better.

3. <u>The third experiment</u>

The preprocessing was done using the following steps:

- Endlines and multiple spaces were removed.
- Numbers have been transformed into letters.
- Stemming and TfidfVectorizer were used on the resulting data.
- TfidfVectorizer parameters:

  TfidfVectorizer(preprocessor=preprocessing_function, tokenizer=tokenize, token_pattern=None, max_features=7000, binary=True)
- The features were scaled using L1 normalization.

The classification was made using linear SVM, 80% of the data was used for training the rest was used for testing.

The SVM was used with the following parameters:

C=10.0, kernel='linear', tol=0.01

The two classifiers were tested on the testing data divided into 5 categories, then into 3 categories an gave the following results:

- 5 categories:
    - Accuracy=62%, F1(weighted)=56%
- 3 categories:
    - Accuracy= 82%, F1(weighted)= 79%

4. <u>The fourth experiment</u>

The preprocessing was done using the following steps:

- Endlines, emoji and multiple spaces were removed.
- Numbers have been transformed into letters.
- Using the Tokenizer class in the keras module, the texts were vectorized, turning them into a sequence of integers by associating a dictionary index with each word in the text.
- In order to have the data of the same length, the chosen length being 300, a padding was added to each representation of the data.

The classification was made using the CNN, 80% of the data was used for training, 10% was used for validation and the rest was used for testing. The CNN model includes an embedding layer with an output dimension of 100, 1d conventional layers with ReLU as activation function and 1d max pooling layers used to reduce the dimensions of the features.

The two classifiers were tested on the testing data divided into 5 categories, then into 3 categories an gave the following results:

- 5 categories:
    - Accuracy=60%, F1(weighted)=57%
- 3 categories:
    - Accuracy= 83.5%, F1(weighted)= 81%

Conclusion

- Fine-grained opinion mining has a low accuracy - approximately 60% - because the trained models cannot distinguish between the categories 'bad' and 'very bad' and the categories 'good' and 'very good'.
- With 3 categories the trained models give a much higher accuracy - approximately 80%. The result reflects reality, since people can more easily categorize a review as being generally 'good' or 'bad', rather than its precise opinion.

Bibliography

Gowtamreddy, P. (2014). *Opinion Mining of Online Customer Reviews*. Available at: https://core.ac.uk/download/pdf/80147109.pdf [1]

Khafajeh, H. (2020). *Opinion Mining: How efficient are Online classification tools?* International Journal of Emerging Trends in Engineering Research, [online] 8. Available at: warse.org/IJETER/static/pdf/file/ijeter46822020.pdf

Razali, N.A.M., Malizan, N.A., Hasbullah, N.A. et al. (2021). *Opinion mining for national security: techniques, domain applications, challenges and research opportunities*. Available at: https://doi.org/10.1186/s40537-021-00536-5 [2]

SearchEnterpriseAI. *What is convolutional neural network?* [online] Available at: https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network

Team, T.A. (2021). *Sentiment Analysis (Opinion Mining) with Python — NLP Tutorial.* [online]. Available at: https://pub.towardsai.net/sentiment-analysis-opinion-mining-with-python-nlp-tutorial-d1f173ca4e3c

www.clarin.eu. *Tools for sentiment analysis and opinion mining*. [online] Available at: https://www.clarin.eu/resource-families/tools-sentiment-analysis-and-opinion-mining