Sentiment Classification on Customer Review Data

Wu, Bi (Vicky)

Artificial Intelligence Final Project

CUNY Graduate Center

Table of Contents

Abstract

Sentiment analysis has long been a focus in the research of text categorization, the task of assigning a label to a piece of text or document. Sentiment analysis aims to identify the positive or negative orientation that a writer expresses towards a subject and it is widely used under business contexts to help assess customers' satisfaction level towards a product or a brand. Sentiment analysis can be done using various methods of training and text processing. This paper focuses on exploring the efficiency of a simple sentiment classification task using multiple natural language processing methods including different word vectorization techniques, and various machine learning models, including Naïve Bayes, KMeans, Logistic Regression, SVM, LDA and RNN, by discussing the application and advantages versus disadvantages with each of these methods.

Keywords: Sentiment Analysis, Natural Language Processing, Artificial Intelligence, Binary Classification, Bag-of-words, TFIDF, Naïve Bayes, KMeans, Logistic Regression, Support Vector Machine, Linear Discriminant Analysis, Recurrent Neural Network

Introduction

Sentiment analysis is one of the fastest growing research areas in artificial intelligence, with not only computer scientists but also a growing number of businesses adapting the practice to help them better understand their audiences and customers. According to a research from the Institute of Software Technology, University of Stuttgart, nearly 7,000 papers of the topic "sentiment analysis" have been published up until Oct. 2016 and 99% of these papers appeared after 2004. As seen in Figure 1, the number of search results for "sentiment analysis" increases dramatically in Google search engine from 2004 to 2016, compared with a derivative topic "customer feedback". (Mika V. Mäntylä1 Daniel Graziotin, 2018)

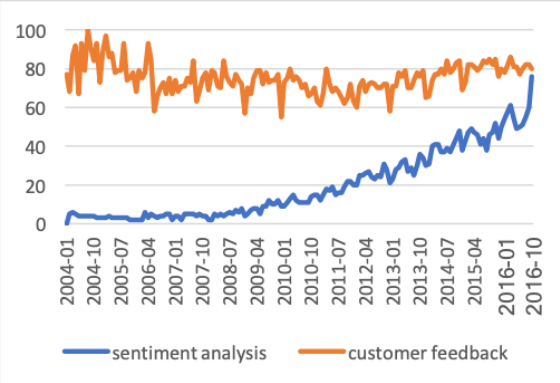

Figure 1

The approach and content of sentiment analysis has also changed drastically over the years. Prior to the era of big data and social media, where now massive amounts of online customer data and public opinions are within free reach of anyone on the Internet, public sentiment studies mainly relied on survey-based primary research methods and the results were often skewed by the opinion of the person studying the matter.

In the mid 90s, computer-based systems started to emerge and started to be reflective in sentiment analysis as well. A paper titled "Elicitation, Assessment, and Pooling of Expert Judgments Using Possibility Theory" published in 1995, used a computer system for expert opinion analysis in the domain industrial safety that allowed for example a pooling of opinions. (S. A. Sandri, 1995) The outbreak of modern sentiment analysis, however, was not until the early 2000s, when researchers started to focus on opinion mining and semantic classification on the sentence or phrase-level. (Mika V. Mäntylä, 2018) The paper, "Opinion mining and sentiment

analysis" published in 2008, was one of the earliest papers discussing modern sentiment analysis and was cited in 2487 times in the years followed. (Lee, 2018)

There are three main existing approaches to perform sentiment analysis: knowledge-based techniques, statistical methods, and hybrid approaches. (Cambria, Schuller, Xia, & Havasi, 2013) Knowledge-based techniques look for unambiguous affect words such as happy and sad in a piece of text to classify text sentiment. (Ortony, Clore, & Collins, 1988) Statistical methods apply machine learning techniques such as support vector machine, Naïve Bayes algorithm, word vectorization, bag-of-words approach for identifying semantic orientation. Hybrid approaches leverage both knowledge-based techniques and machine learning elements in order to better detect semantics that are expressed in a subtle manner. (Cambria & Hussain, Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis, 2015)

The methodology discussed in this paper focuses on statistical model application using open-source machine learning packages from Scikit-learn in Python. These models include Multinomial Naïve Bayes, KMeans Clustering, Logistic Regression, Support Vector Machine, Linear Discriminant Analysis as well as deep learning model Recurrent Neural Network. In addition, the text processing approaches discussed in this paper is heavily relied on the bag-of-words model, which disregarded grammar, word order, word context and other nuances that could have affected the paragraph-level sentiment being analyzed.

Research Methods

2.1 Data

The research is done by using a real commercial customer review dataset for an online clothing store, which can be accessed through https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews. The dataset includes 23486 rows and 10 feature variables. A snapshot of the first 5 rows is shown in Figure 2. Each row corresponds to a customer review, and a description for each feature variable can be found on the website.

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	Tops	Blouses

Figure 2

The classification task performed in this research utilized only two of the ten feature variables in the dataset: “Review Text” and “Recommended IND”. The “Review Text” variable is used as the predictor while the “Recommended IND” variable is used as the labels. A description of the two variables is written below:

- Review Text: String variable for the customer review body.
- Recommended IND: Binary variable stating whether the customer recommends the product, where 1 is recommended, 0 is not recommended.

The goal of the task is to classify each review into either “recommended” or “not recommended” category based on the sentiment expressed in the review body. One can confidently associate the level of positivity expressed in a customer review with the likelihood of the customer recommending the product to other people. In fact, sentiment analysis has proven to be an effective technique in building recommender systems. (Noah, 2018)

2.2 Text Processing

To be able to use statistical models for our classification task, text data needs to be converted into numeric values. The following steps were taken to process the review body text:

- Data sampling and shuffling
- Word segmentation (Tokenization)
- Remove punctuation and numbers
- Remove words of less semantic meaning
- Word lemmatization
- Word vectorization

The original data contains 23486 rows with nearly 80% of the rows in the recommended category and only 20% in the not recommended category. To be able to effectively train our

models later on, 1000 entries from each category were randomly selected and then shuffled to be used for the classification task. Each of the 2000 reviews was then tokenized into a list of characters, with punctuation, numbers, and words of less semantic meaning removed from the list. Afterwards, for each word in the word list, the original form of the word was returned, thus drastically reducing the total number of unique words in the entire 2000 reviews. An example of a review tokenized into a list of words is shown in figure 3 and figure 4.

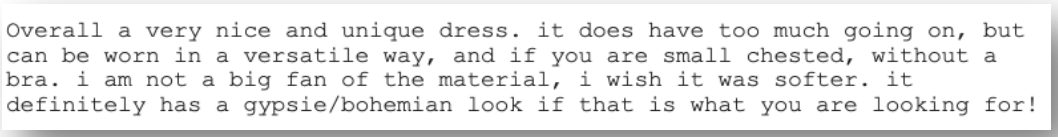

Figure 3

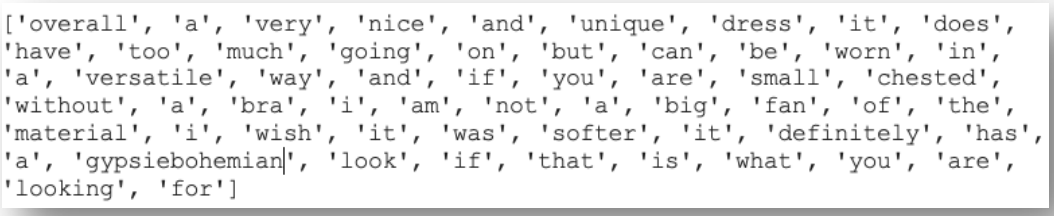

Figure 4

In the last step of word vectorization, two approaches were being used to map words from vocabulary to a corresponding vector of real numbers: Bag-of-words approach and TF-IDF score. In the bag-of-words model, every customer review is represented as a bag of all the words appeared in the entire corpus, disregarding grammar, word order but keeping the count of each word within the customer review. In the TF-IDF score approach, a numeric statistic is being assigned to each word in each customer review to decide how import that word is to the review itself and to the entire document. In the TF-IDF vectors, each word is represented as a feature and the scores are assigned as the values, as shown in Figure 5.

add	added	addict	addition	...	xspwas	xtr	xx	xxl	xxs	xxsit	xxsmall	xxsp	yak	yarn	yay	yeah	year	yellow
0.212044	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00000
0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00000
0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00000
0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.16981
0.000000	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.00000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 5

Because the TF-IDF score matrix is a sparse matrix with large amounts of values being 0, and most values less than 0.5, a data scaler called MinMaxScaler from sklearn is used to scale the features values in between 0 and 1.

The 2000 customer reviews were then split into a training set and a testing set, with 80% of the data – 1600 reviews in the training set and 20% - 400 reviews in the testing set.

2.3 Multinomial Naïve Bayes

Multinomial Naive Bayes model is widely used for the categorization of the text and it assumes class conditional independence, in that the effect of each word on a given class is independent from the effect of other words. The model is developed based on the mathematic equation below where a customer review x is being assigned to class c where c = argmaxP(Ck | x).

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

In our case, the denominator of the equation, which is the probability of the evidence is ignored, assuming they are the same for each customer review. The priors, p(Ck), are calculated based on the number of observations appeared in each class in the training set. The likelihoods of a customer review x appear in a given class Ck is calculated based on the number of times each word in the customer review occur in the different classes in the training set.

2.4 KMeans Clustering

KMeans clustering works to partition n observations into k clusters in which each observation belongs to a cluster that has the shortest distance from the cluster centroid to the observation. In our case, the number of clusters is set to be 2 as we only have 2 classes, each customer review (observation) is transformed into a vector or bag-of-words count or a vector of TF-IDF scores, then the Euclidean distances between each vector and the 2 cluster centroids are calculated. Given a set of vectorized customer reviews (x1, x2, ..., xn), the objective is to

minimize the within-cluster sum of squares (variance), as shown in the mathematic formulas below:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

When used in combination with PCA and number of components set to 2, the clusters can be plotted on a two-dimensional graph. Below is a visualization of what the two clusters look like for the customer review data using PCA transformed two-dimensional TF-IDF scores, with x and y axis representing principal components 1 and 2.

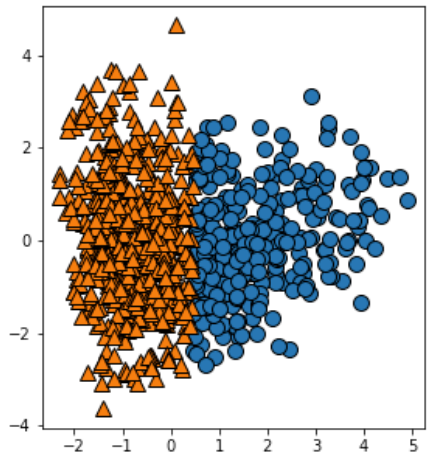

Figure 6

2.5 Logistic Regression

Logistics regression is also a widely used supervised machine learning technique for binary classification tasks. It assumes a linear relationship between the predictor variables (x) and the log odds of the event that Y=1 (In our case, Y=1 is the event of a predictor being classified into the “recommended” class){\displaystyle Y=1}. Each vectorized customer review is treated as a predictor and the probability of the predictor falling into class 1 (Y=1) is calculated, if the probability is larger than .5, the customer review will be classified as “recommended”, if the probability if in between 0 and .5, the customer review will be classified as ‘not recommended”.

2.6 Support Vector Machine

SVM is a supervised machine learning algorithm that can be used for both classification and regression purposes. SVM does classification by finding a hyperplane that best separates the classes in a n-dimensional space. In our case, the vectors of bag-of-words scores or TF-IDF

scores are fed into the algorithm and transformed by mathematical functions like sigmoid, polynomial, radio basis function so that they can be better separated.

2.7 Principal Component Analysis (PCA)

Principal component analysis is commonly used for dimension reduction when the data is high-dimensional. In our case, PCA is applied on the TF_IDF score vectors in combination with Naïve Bayes, KMeans, SVM and Logistic Regression models, before fitting the vectors into the models. The results are then compared with using the TF-IDF vectors without dimension reduction. The original TF-IDF score vectors has 4369 dimensions, PCA was applied with the number of components set to keep a 95% summative variance of all individual components, thus reducing the number of dimensions to 1316.

2.8 Linear Discriminant Analysis (LDA)

Linear discriminant analysis is closely related to PCA in that they both look for linear combinations of variables that best explains the data. (Martinez & Kak, 2001) The difference is that LDA attempts to model the differences between classes of the data while PCA does not take class into account. LDA is also commonly used for dimension reduction purposes before classification, but it can also be used as a linear classifier itself. In our case, LDA works perfectly as a classifier in separating the two classes, when used in combination with TF-IDF scores. The number of dimensions of the TF-IDF vectors, are reduced to 1 when there are only 2 classes. The transformed 1-dimensional TF-IDF scores of the 1600 data points in the training set can be visualized as below:

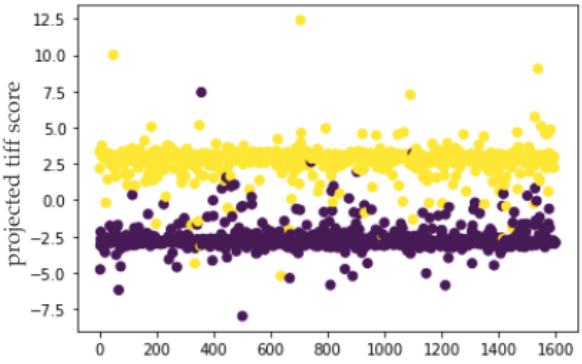

Figure 7

2.9 Recurrent Neural Network

In our case of text classification, a recurrent neural network is used with LSTM to perform the task in Keras. During the text processing step, instead of using conventional word

vectorization methods like bag-of-words and TF-IDF scores, each word in each review is mapped with an integer that ranks the frequency of all the words in the entire training set. For example, the word appeared the most times is assigned to an integer value of 1, and the word appeared the second most time is assigned to 2 and so forth. Each review is then padded to the same length of sequence (using the maximum length of sequence of all the reviews) by adding integer 0 in the front. The model is trained with the sigmoid activation function and the binary cross-entropy loss function.

Results

With 1600 customers in the training data and 200 customer reviews in the testing data, the accuracies of the classifiers are listed below:

- Naïve Bayes
    - Built from scratch with bag-of-words: 85.75%
    - Multinomial Naïve Bayes from Scikit-learn used with bag-of-words: 82%
    - Multinomial Naïve Bayes from Scikit-learn used with TF-IDF scores: 81% (not recommended as Multinomial Naïve Bayes is suited for discrete features)
- KMeans Clustering
    - Used with bag-of-words: 53%
    - Used with TF-IDF scores: 49.5%
    - Used with TF-IDF scores and PCA: 66.25%
- Logistic Regression
    - Without regularization, used with bag-of-words: 79.2%
    - Without regularization, used with TF-IDF scores: 79.2%
    - Without regularization, used with TF-IDF scores and PCA: 78.75%
    - With L2 penalty, used with bag-of-words: 81%
    - With L2 penalty, used with TF-IDF scores: 85.2%
    - With L2 penalty, used with TF-IDF scores and PCA: 83%
- SVM
    - Used with bag-of-words: 81.75%

    - Used with TF-IDF scores: 85.25%
    - Used with TF-IDF scores and PCA: 84%
- LDA
    - Used with bag-of-words: 64.5%
    - Used with TF-IDF 98%
- Recurrent Neural Network
    - Used with frequency rank integer mapping: 81%

Conclusions

Among the above models, LDA used in combination with TF-IDF scores yielded the highest accuracy at 98%. Naïve Bayes with bag-of-words approach, logistic regression with L2 penalty on TF-IDF scores and SVM with TF-IDF scores produced similar high accuracy at around 85%. KMeans clustering proved to be less efficient at this classification task, which is likely owing to the fact that KMeans clustering is an unsupervised machine learning model, often used in topic discovering in the context of a multi-topic piece of text. Given that the customer review data is revolved around the single topic of clothing review, the two clusters produced by KMeans are not well separated from each other, which was seen in Figure 5.

Between the two word vectorization methods, TF-IDF scores produced a better result when used with more complicated models such as Logistic regression with L2 penalty, SVM, and LDA. However, bag-of-words approach exceled when applied with the simple Multinomial Naïve Bayes model that I built from scratch, ignoring the probability of evidence, and gave the second highest accuracy score next to LDA.

Principal component analysis had no or slightly negative impact on the classification accuracy of an SVM or a logistic regression model when used with TF-IDF scores. However, it does help to improve the accuracy of the KMeans clustering model, making it less confusing for the model to distinguish between clusters. The possible explanation for the limited impact of PCA in our case is that the correlation coefficient between TF-IDF features vectors in our data is extremely small (as shown in Figure 7), meaning the features are not strongly correlated with each other, thus making it pointless to use PCA to reduce dimensions.

```
Pick two random features
The coefficient between feature 3309 and feature 3310 is -0.005031184416482022
The coefficient between feature 3036 and feature 3037 is -0.000669614647399663
The coefficient between feature 1689 and feature 1690 is -0.0011640510850932598
The coefficient between feature 3691 and feature 3692 is -0.0011134162672356985
The coefficient between feature 358 and feature 359 is -0.010148609310979595
The coefficient between feature 1036 and feature 1037 is -0.006486090513358903
The coefficient between feature 3970 and feature 3971 is -0.0005002501250625397
The coefficient between feature 2577 and feature 2578 is -0.0008663892219153888
The coefficient between feature 1032 and feature 1033 is -0.0006936710434172047
The coefficient between feature 1956 and feature 1957 is -0.0019148523122171487
...
```

Figure 8 Correlation between two features

Future Research

Future research on the topic will be approached from the following directions:

- Take into consideration grammar and word order in feature extraction. Instead of tokenizing words one at a time, two words or multiple words where a phrase is formed can be tokenized together.
- Apply different text vectorization methods like Word2Vec to convert text to numeric values.
- Try different data scaling/normalization methods at different steps, including before and after dimension reduction.
- Tune model parameters to achieve better accuracy scores.

Bibliography

Cambria, E., & Hussain, A. (2015). *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis.* Springer. Retrieved from Sentiment Analysis: https://en.wikipedia.org/wiki/Sentiment_analysis#References

Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 15-21.

Lee, B. P. (2018). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 1-135.

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 228-233.

Mika V. Mäntylä, D. G. (2018). The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers. *Computer Science Review, Volume 27*, 26-32.

Mika V. Mäntylä1 Daniel Graziotin, M. K. (2018). The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers. *Computer Science Review*, 26-32.

Noah, N. A. (2018). Sentiment-Based Model for Recommender Systems. *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, (pp. 1-6). Kota Kinabalu.

Ortony, A., Clore, G., & Collins, A. (1988). *The Cognitive Structure of Emotions.* London: Cambridge Univ. Press.

S. A. Sandri, D. D. (1995). Elicitation, assessment, and pooling of expert judgments using possibility theory. *IEEE transactions on fuzzy systems*, 313-335.