Feature extraction and classification of movie reviews

Mtetwa, Nhamo; Awukam, Awukam Ojang; Yousefi, Mehdi

Published in:
2018 5th International Conference on Soft Computing &amp; Machine Intelligence (ISCMI)

DOI:
10.1109/ISCMI.2018.8703235

Publication date:
2019

Document Version
Peer reviewed version

Link to publication in ResearchOnline

Citation for published version (Harvard):
Mtetwa, N, Awukam, AO & Yousefi, M 2019, Feature extraction and classification of movie reviews. in *2018 5th International Conference on Soft Computing & Machine Intelligence (ISCMI).* IEEE, pp. 67-71. https://doi.org/10.1109/ISCMI.2018.8703235

Feature Extraction and Classification of Movie Reviews

Nhamo Mtetwa, Awukam Ojang Awukam and Mehdi Yousefi
CCIS Department, Glasgow Caledonian University
Glasgow, United Kingdom
E-mail: {nhamoinesu.mtetwa, mehdi.yousefi}@gcu.ac.uk
oawuka200@caledonian.ac.uk

Abstract— **Sentiment analysis identifies a user's attitude towards a service, a topic or an event and it is very useful for companies that receive many written reviews of their services. We investigate the effect of feature extraction techniques on supervised machine learning classifiers using four different performance metrics using a publicly available movie review dataset. Our objective is to explore different classification algorithms as well as utilizing diverse feature extractors and compare outcomes and finally select the trio of feature extraction technique, classification algorithm and performance metric with the best result for the movie review classification use case.**

Keywords: feature extraction; machine learning; sentiment analysis; support vector machine; random forest

I. Introduction

Sentiment analysis could be defined as: *The process of algorithmically identifying and categorizing opinions expressed in text to determine the user's attitude toward the subject of an expressed opinion.* In other words: we can analyse if people at large generally like or dislike a service or product and in our use case whether they like or dislike a movie. We live in a digital world dominated by the World Wide Web which is growing at an exponential rate [1]. A big component of the web is people's reviews or opinions of digital services. Mining people's opinions is becoming a source of competitive intelligence and advantage for any business but more so for online businesses. Sentiment analysis is an active area of research in view of its broad application in domains such as advertising organizations, as they investigate their services and products, to institute brand improvement [2].

Movie review websites like IMDB, Amazon Prime and Netflix enable users or clients to submit reviews in the form of ratings. This helps users express their thoughts and opinions about the movies. These remarks can be utilized to produce helpful data that describes its contents, giving other users a chance to find out about the sentiment polarity of those reviews. Reviews may be in thousands and even thousands of thousands to read. It will be time-consuming to scan through all these reviews to conclude whether to purchase a movie or watch it. Likewise, just reading a few reviews will not be sufficient to settle a movie's viewpoints. So it will be useful for both movie producers and potential viewers if movie reviews could be automatically mined [3] and summarized with defined generalized information. In this paper, we classify movie reviews into positive or negative polarity, by applying machine learning and natural language processing for opinion mining and sentiment analysis [4].

The rest of the paper is organized as follows: in section II, we review some work on feature extraction from textual data, machine learning classifiers and machine learning performance metrics. In section III, the details of the methodology for the practical implementation are discussed. In section IV, we present the results. Finally, in section V, the paper is concluded.

II. Literature Review

The first research on sentiment analysis was the measurement of public opinions after and during WWII, which their motivation in nature was political [5], [6]. Modernized sentiment analysis grew in the mid-2000, and it centred around product reviews accessible on the Web [7]. From that point forward, the utilization of sentiment analysis has achieved a lot of success, for example, the prediction of money related markets [8] and reactions to terrorist attacks [9]. Moreover, research combining natural language processing and sentiment analysis has addressed numerous issues that add to the relevance of sentiment analysis, for example, irony detection [10] as well as multi-lingual support [11]. Additionally, regarding emotions, endeavours are progressing from basic polarity identification to more complicated subtleties of emotions and separating negative emotions, for example, grief and anger [12]. Critically looking at sentiment analysis, it is all about opinion polarity, which literally means either someone's opinion is negative, neutral or positive towards something [7].

Machine learning and natural language processing (NLP) are the main tools that are dominating sentiment analysis. There is a lot of focus on machine learning but not enough focus on what happens before the machine-learning step. Machine learning models, such as neural networks, decision trees, random forests and gradient boosting machines accept a feature vector and provide a prediction [13]. These models learn in a supervised fashion where a set of feature vectors with expected output is provided [14]. Feature extraction and selection have become the focus of much research in areas of application for which datasets with tens or hundreds of thousands of variables are available [15]. These areas include text processing of Internet documents, gene expression array analysis, and combinatorial chemistry [16]. Feature selection is the process of selecting a subset of relevant features (variables, predictors). The objective of feature selection is three-fold: improving the prediction performance of the predictors/classifiers, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data [17]. In this paper, feature extraction and selection for use in a movie review classification model is discussed. The paper discusses three different feature extraction methods and three different classification algorithms. The target application for this exercise is classification of movie reviews as either positive or negative.

A *feature* is a numeric representation of raw data [18]. There are many ways to turn raw data into numeric measurements. Basically, features must derive from the type of data that is available. Perhaps less obvious is the fact that they are also tied to the model; some models are more appropriate for some types of features and vice versa. The right features are relevant to the task at hand and should be easy for the model to ingest. *Feature selection* is the process of formulating the most appropriate features given the data, the model, and the task [17], [19]. The NIPS 2003 Feature Selection Challenge offered a great testbed for evaluating feature selection algorithms on datasets with a very large number of features as well as relatively few training examples [20].

The number of available features that can be linked to a document or text is huge. These features are associated with the syntax and semantics of the text. One primary challenge in operationalizing sentiment analysis is to identify the smallest set of features before classifying the sentiments as positive or negative. Not considering this challenge may cause deterioration in the classification rate especially when many redundant features are kept in the dataset. These redundant features increase the search space for the classification algorithm.

Many approaches to extracting features from text have been designed to solve specific problems and operate in ad-hoc domains [21]. Other approaches, instead, heavily reuse techniques and algorithms developed in the field of information extraction. Various machine-learning applications are usually overwhelmed by a large number of features.

Sentiment analysis is the task of classifying text or documents according to their sentiment orientation. Before classification of text sentiment, the plain text documents need to be transformed into features for machine learning classification of the sentiments as positive or negative. This step is known as feature extraction. Feature extraction produces text representations that are enriched with information in order to have better classification results. This paper presents a comparison of three different feature extraction techniques with three classification techniques to achieve viable sentiment analysis. The feature extraction techniques used are *word count vector* [22], *term frequency–inverse document frequency* (TF-IDF) [23], [24] and *bi-grams* [25].

TF-IDF is a statistic that reflects how important a word is to a specific document relative to all of the words in a collection of documents (the corpus) [26]. The TF-IDF value increases proportionally to the number of times that word appears in the document, but is offset by the frequency of the word in the corpus [27].

Another way to represent a text document is to count the instances of every word in the document. Articles can then be compared based on how similar their *word count vectors* are [28].

The final feature extraction is *bi-grams*, which is a subset of so called n-grams. n-grams are basically a set of occurring words within given windows so when

- n=1 it is Unigram
- n=2 it is bigram
- n=3 it is trigram and so on

The basic point of n-grams is that they capture the language structure from the statistical point of view, like what letter or word is likely to follow the given one. The longer the n-gram (the higher the *n*), the more context one has to work with.

One of our objectives is to explore different avenues regarding distinctive algorithms as well as utilizing diverse feature extractors and compare outcomes or results and finally select the algorithm with the best accuracy for movie review classification. As for machine learning classifiers, we picked three popular algorithms: Multinomial Naive Bayes, Random Forest, and Support Vector Machine (SVM).

A. *Multinomial Naïve Bayes Classifier*

Multinomial Naïve Bayes is basically multiclass Naïve Bayes. In multinomial naive Bayes, the features are assumed to be generated from a simple multinomial distribution [29]. The multinomial distribution describes the probability of observing counts among a number of categories, and thus multinomial naive Bayes is most appropriate for features that represent counts or count rates which suits our movie review use case. One place where multinomial naive Bayes is often used is in text classification, where the features are related to word counts or frequencies within the documents to be classified.

B. *Random Forest Classifier*

Random Forest belongs to the family of decision trees, which is used for solving classification as well as regression problems. Decision trees work by separating the dataset into incrementally created small subdivisions. Despite the fact that decision trees are straightforward and have demonstrated great results in classification analysis, they are susceptible to overfitting the data. Even though they encounter the issue of overfitting in a learning scenario, they always find a way out [30], in extracting genuine learning knowledge from the presented data. This notion that multiple overfitting estimators can be combined to reduce the effect of overfitting is what underlies an ensemble method called *bagging*. Bagging makes use of an ensemble (a grab bag, perhaps) of parallel estimators, each of which overfits the data, and averages the results to find a better classification. An ensemble of randomized decision trees is known as a *random forest.*

C. Support Vector Machine (SVM)

SVM is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. For categorical variables, a dummy variable is created with case values as either 0 or 1. To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function.

D. Performance Metrics

With so many supervised machine learning classifiers available there is need for a way to evaluate their classification capability. In this paper we consider four performance metrics: accuracy, precision, recall and f1-score.

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observations to the total observations. One may think that, if they have high accuracy then their model is best. Yes, accuracy is a great measure but only when one has symmetric datasets where values of false positive and false negatives are almost the same. Therefore, one has to look at other parameters to evaluate the performance of their model.

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answers is: of all the reviews that are classified as positive, how many are actually positive? High precision relates to the low false positive rate.

Recall (sensitivity) is the ability of a model to find all the relevant cases within a dataset. The precise definition of recall is the number of true positives divided by the number of true positives plus the number of false negatives. The question recall answers is: Of all the reviews that are truly positive, how many did the model identify?

The *F1-score* is the harmonic mean of precision and recall taking both metrics into account. F1-score is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both precision and recall.

III. METHOD

The methodology followed in this work is summarised in Fig. 1.

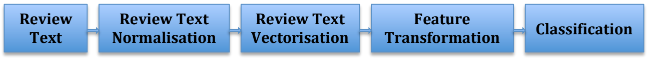

Fig. 1. Flowchart of the methodology followed

E. Dataset

For the data, we use Stanford University's ACL IMDB movie review dataset [31]. This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. It provides a set of 25,000 movie reviews for training, and 25,000 reviews for testing. There is additional unlabelled data for use as well.

F. Processing Pipeline

The raw data is loaded and pre-processed. Pre-processing is one of the key components in most text mining algorithms. The pre-processing step consists of tasks such as tokenization, filtering, lemmatization and stemming which we collectively call normalisation. Text normalization is the process of transforming text into a single canonical form suitable for feature extraction. Normalizing text before processing it allows for separation of concerns since the input is guaranteed to be consistent before operations are performed on it. After normalising the move reviews data we create features using TF-IDF, bi-grams and word count vector vectorisation techniques. After suitably transforming the features for each of the three machine learning algorithms we train the models using the training dataset and then test using the testing dataset. We evaluate each trained model using four performance metrics namely accuracy, recall, precision and F1-score. The whole pipeline is implemented in a python jupyter notebook supported by the sk-learn python machine learning library.

IV. RESULTS

TABLE 1 summaries all the results of the three feature extraction techniques applied to three machine learning classifiers being scored against each other using four performance metrics. Each score or metric is the average for positive and negative classifications scores for that model.

The lowest result is for bi-grams and word count vector feature extraction applied to random forest. The best result is TF-IDF applied to SVM and bi-grams applied to Multinomial Naïve Bayes.

Table 1: List of features used

Features		Counter Vector	TF-IDF	Bi-grams
SVM	Precision	0.86	0.88	0.87
	Recall	0.86	0.88	0.87
	F1-score	0.86	0.86	0.87
	Accuracy	0.86	0.86	0.87
Multinomial NB	Precision	0.85	0.86	0.88
	Recall	0.85	0.86	0.88
	F1-score	0.85	0.86	0.88
	Accuracy	0.85	0.86	0.88
Random Forest	Precision	0.86	0.85	0.84
	Recall	0.84	0.85	0.84
	F1-score	0.85	0.85	0.84
	Accuracy	0.85	0.85	0.84

V. Conclusion

The number of features in a model is important. If there are not enough informative features, then the model will be unable to fulfil its ultimate task. If there are too many features, or if most of them are irrelevant, then the model could go awry in the training process which impacts the model's performance [32].

Features and models sit between raw data and the desired insights. In a machine learning workflow, we pick not only the model, but also the features. This is a double-jointed lever, and the choice of one affects the other. Good features make the subsequent modelling step easy and the resulting model more capable of achieving the desired task in a timely manner. Bad features may require a much more complicated model to achieve the same level of performance. The more thoughtful input features one has, the better the accuracy and efficiency of the model.

In our experiment we consider the relation between different supervised classification techniques as well as the relation between different choices of the feature extractor. The use of an appropriate metric to score the models especially in situations where the cost of positive and negative reviews is different is also important. Precision is a good measure when the cost of false positives is high. For instance, email spam detection. In email spam detection, a false positive means that an email that is non-spam (actual negative) has been identified as spam (predicted spam). The email user might lose important emails if the precision is not high for the spam detection model. Recall is the model metric of choice when there is a high cost associated with false negatives, which could be the case in social media sentiment analysis.

The natural way to extend this work is to consider both feature extraction and classification techniques which take into consideration both the syntactic and semantic structure of the reviews like deep learning and recurrent neural networks to take care of reviews which mix both negative and positive sentiments in the same review [33]. The other future consideration is to measure the cost of negative and positive reviews and create appropriate weightings to score the models.

References

[1] B. A. Huberman and L. A. Adamic, "Internet: growth dynamics of the world-wide web," Nature, vol. 401, p. 131, 1999.

[2] P. V. Rajeev and V. S. Rekha, "Recommending products to customers using opinion mining of online product reviews and features," in Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on, 2015.

[3] M. Hu and B. Liu, "Mining opinion features in customer reviews," in AAAI, 2004.

[4] A. Kumar and T. M. Sebastian, "Sentiment analysis: A perspective on its past, present and future," International Journal of Intelligent Systems and Applications, vol. 4, pp. 1-14, 2012.

[5] R. Stagner, "The cross-out technique as a method in public opinion analysis," The Journal of Social Psychology, vol. 11, pp. 79-90, 1940.

[6] A. L. Knutson, "Japanese opinion surveys: the special need and the special difficulties," Public Opinion Quarterly, vol. 9, pp. 313-319, 1945.

[7] K. Dave, S. Lawrence and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in Proceedings of the 12th international conference on World Wide Web, 2003.

[8] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah and D. C. L. Ngo, "Text mining for market prediction: A systematic review," Expert Systems with Applications, vol. 41, pp. 7653-7670, 2014.

[9] P. Burnap, M. L. Williams, L. Sloan, O. Rana, W. Housley, A. Edwards, V. Knight, R. Procter and A. Voss, "Tweeting the terror: modelling the social media reaction to the Woolwich terrorist attack," Social Network Analysis and Mining, vol. 4, p. 206, 2014.

[10] A. Reyes and P. Rosso, "On the difficulty of automatically detecting irony: beyond a simple case of negation," Knowledge and Information Systems, vol. 40, pp. 595-614, 2014.

[11] A. Hogenboom, B. Heerschop, F. Frasincar, U. Kaymak and F. de Jong, "Multi-lingual support for lexicon-based sentiment analysis guided by semantics," Decision support systems, vol. 62, pp. 43-53, 2014.

[12] E. Cambria, P. Gastaldo, F. Bisio and R. Zunino, "An ELM-based model for affective analogical reasoning," Neurocomputing, vol. 149, pp. 443-455, 2015.

[13] Y. Saeys, T. Abeel and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2008.

[14] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," in SoutheastCon, 2016, 2016.

[15] K. Kira and L. A. Rendell, "A practical approach to feature selection," in Machine Learning Proceedings 1992, Elsevier, 1992, pp. 249-256.

[16] J. Miao and L. Niu, "A survey on feature selection," Procedia Computer Science, vol. 91, pp. 919-926, 2016.

[17] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of machine learning research, vol. 3, pp. 1157-1182, 2003.

[18] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Computers \& Electrical Engineering, vol. 40, pp. 16-28, 2014.

[19] N. Mtetwa, M. Yousefi and V. Reddy, "Feature selection for an SVM based webpage classifier," in Soft Computing \& Machine Intelligence (ISCMI), 2017 IEEE 4th International Conference on, 2017.

[20] I. Guyon, S. Gunn, A. B. Hur and G. Dror, "Design and Analysis of the NIPS2003 Challenge," in Feature Extraction, Springer, 2006, pp. 237-263.

[21] E. Ferrara, P. De Meo, G. Fiumara and R. Baumgartner, "Web data extraction, applications and techniques: A survey," Knowledge-based

systems, vol. 70, pp. 301-323, 2014.

[22] D. Jurafsky and J. H. Martin, Speech and language processing, vol. 3, Pearson London, 2014.

[23] M. Qasem, R. Thulasiram and P. Thulasiram, "Twitter sentiment classification using machine learning techniques for stock markets," in Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on, 2015.

[24] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, 2002.

[25] A. Z. Broder, S. C. Glassman, M. S. Manasse and G. Zweig, "Syntactic clustering of the web," Computer Networks and ISDN Systems, vol. 29, pp. 1157-1166, 1997.

[26] A. Tripathy, A. Agrawal and S. K. Rath, "Classification of sentimental reviews using machine learning techniques," Procedia Computer Science, vol. 57, pp. 821-829, 2015.

[27] P. Tiwari, B. K. Mishra, S. Kumar and V. Kumar, "Implementation of n-gram methodology for rotten tomatoes review dataset sentiment analysis," International Journal of Knowledge Discovery in Bioinformatics (IJKDB), vol. 7, pp. 30-41, 2017.

[28] M. B.-H. A. a. E. M. Birjali, "Prediction of Suicidal Ideation in Twitter Data using Machine Learning algorithms," in International Arab Conference on Information Technology (ACIT'2016), 2016.

[29] R. Rana and V. Kolhe, "Analysis of Students Emotion for Twitter Data using Na\'ive Bayes and Non Linear Support Vector Machine Approachs," International Journal on Recent and Innovation Trends in Computing and Communication. ISSN, pp. 2321-8169, 2015.

[30] S. Buschjäger and K. Morik, "Decision Tree and Random Forest Implementations for Fast Filtering of Sensor Data," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, pp. 209-222, 2018.

[31] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and C. Potts, "Learning word vectors for sentiment analysis," in Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1, 2011.

[32] O. Chapelle and S. S. Keerthi, "Multi-class feature selection with support vector machines," in Proceedings of the American statistical association, 2008.

[33] H. K. Dam, T. Tran, T. Pham, S. W. Ng, J. Grundy and A. Ghose, "Automatic feature learning for vulnerability prediction," arXiv preprint arXiv:1708.02368, 2017.