Report:
IMDb Sentiment Analysis

1 Introduction

Sentiments analysis is described as a text analysis tool that determines the opinion, attitude and subjectivity of the reviews. In a collection of text documents containing opinions, each comment (or review) can be classified into either negative or positive attitude category [1]. As the benchmark, a publicly available dataset of movie reviews from IMDb was obtained which contains 25000 labeled movie reviews for training and 25000 unlabeled reviews for testing [2]. In this project, different models were trained and evaluated on the 25000 reviews in the training set, while the other 25000 were used to obtain the accuracy of our final model on the unseen examples. The developed models follow three main blocks including preprocessing, feature selection and classification.

First, we took some preprocessing steps to clean the raw text before entering the feature extraction phase. To do so, punctuation removal, stop-word removal and normalization were performed respectively in the preprocessing phase. In the next step, the clean data was fed into the feature selection module where several features were extracted from the text such as N-grams (N = 1,2,3), TF*IDF, Bing Liu's lexicon[17] and VADER lexicon[18]. During our experiments we noticed the effect of feature dimension reduction on the performance of our models. Using features such as bigrams and trigrams considerably increased the feature matrix dimensions. We realized some of these features were insignificant and noisy that decreased the accuracy of the models. To improve and accelerate the performance, we used Chi-squared dimension reduction modules from Sklearn package[3] to reduce the number of features from approximately 5M to 3M and less, based on the Chi-squared score. Finally, the extracted features were fed to different classifiers such as BNB, MNB, SVM, LR, NBSVM and two variations of ensemble learning methods.

There were some hyper parameters in preprocessing step (e.g. removing or leaving the stop-words), feature selection methods (e.g. N in N-gram, different forms of features from lexicon sentiment collections) and classifiers (e.g. kernel and the value of C in SVM) which provided degree of freedom for selecting the final model. We used the grid search technique to obtain the optimal value for each of the hyper parameters of the different models. These optimal values are supposed to maximize an objective function which is defined as the accuracy of our model on the validation data. In order to propose a robust model with acceptable generalization to the unseen data we used K-fold cross-validation technique on the training data. Among all the classifiers, linear SVM with large values for C ($1e^3$ resulted to the highest accuracy on the training and the test set (90.57 %). The features we used to train our SVM model were N-gram(N=1,2,3) and TF*IDF. We then selected top 3M features with the highest Chi-squared score and train the classifier. We surprisingly observed that Lemmatization and stop-words removal decreased our accuracy by 2 %.

2 Related work

As mentioned above, the task of sentiment analysis includes two main parts of feature selection and classification. The two main methods that are extensively used in the literature for extracting features from the reviews are statistical based and lexicon based approaches [4]. Several methods like term frequency (TF), inverse document frequency (IDF) and term frequency-inverse document frequency (TF-IDF) [5] are used in the statistical feature extraction methods [6]. On the other hand, in the lexicon based methods [7], features are extracted by using a lexicon dictionary, finding a specific pattern among the words, etc. In [8], an improvement in the accuracy of the model was observed by using both the lexicon based and the statistical based methods. Those features, which do not have significant effect on the classification, can be eliminated using feature reduction methods [9]. In [10], authors conducted sentiment classification of the reviews by applying classifiers such as SVM, NB and Maximum Entropy along with using n-grams approaches and bag of words (BOW) as features. The accuracy of 82.9 % was achieved in their study on the movie reviews. Another sentiment analysis study [11], achieved higher performance using unigram features compared to other features. A novel method was used by modeling the documents as graphs with the sentences as the nodes and the association score between them as the edges of the graph [12]. They reported 89 % accuracy which was a significant improvement compared to the previous studies. A discriminative machine learning classifier was used in [13] and the accuracy of 90 % was reported on the online electronic product reviews (320k reviews). Authors in [14] showed that the simple NB and SVM (NBSVM) variants outperformed most of the published studies on several sentiment analysis datasets. In [14], by applying NBSVM on the IMDb database, the accuracy of 91.22 % was obtained. Another interesting approach for sentiment analysis is deep learning which can result in high prediction accuracy (higher than 95 %). A comprehensive review on using deep learning in sentiment analysis can be found in [15].

3 Dataset and setup

The large IMDb movie review dataset [16] contains a collection of 50,000 movie reviews from the Internet Movie Database website along with their associated sentiment polarity labels. The entire collection is highly polarized, meaning that a negative review has a score ≤ 4 out of 10 and a positive review has a score ≥ 7. The dataset is split into two 25k training and test sets.

3.1 Pre-processing Procedure

The entries are in the form of raw text, meaning they contain several attributes which do not play a significant role in the training process and thus, some text preprocessing is applied on them in order to optimize the learning process. In this work, this preprocessing is done in the following order:

1. **Punctuation removal:** all punctuation marks are removed from each entry, some of them being replaced with space.

2. **Stop-word removal:** stop-words are the very common words like ‘if’, ‘but’, ‘we’, ‘he’, ‘she’, and ‘they’. These words can be removed without changing the semantics of a text and doing so, often improves the performance of a model

3. **Normalization:** a common step in which different forms of a given word are converted into one. Two existing methods include Stemming and Lemmatization. Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even

if the stem itself is not a valid word in the Language. Lemmatization, unlike Stemming, reduces the inflected words properly ensuring that the root word belongs to the language. In this work, we choose the latter as the method for normalization.

Last two steps were applied using built-in functions of the NLTK Python library.

3.2 Text Features

After analyzing the dataset, we extracted several different features and examined them by feeding those features to different models. We used different orders of N-grams (i.e. N = 1,2,3) as well as TF*IDF features, which both are popular in text processing. All of these features were normalized at the end before training the classifier.

Additionally we extracted both binary and frequency sentiment features based on the Bing Liu's "Opinion Lexicon" collection [17]. There are total of 6789 sentiment words (2006 positive sentiment words and 4783 negative sentiment words) in Bing Liu's collection. The one-hot-encoded binary features were designed based on the existence of each of the positive/negative words in each entry (review) of our dataset and the frequency version represented the number of times each of the sentiment words was repeated in each review. Since the lexicon collection is unbalanced and has different number of words in the positive and negative category, we also designed features that used same number of words from each category by randomly selecting 2006 words from negative sentiment set.

Furthermore we explored the Valance Aware Dictionary and sEntiment Reasoner (VADER) [18] lexicon. VADER is an open-source (under the MIL License) sentiment lexicon based on the sentiments expressed in social media. For each comments, it produces four sentiment scores (i.e. negative, positive, neutral and compound). To calculate the first three scores, the module extracts all the words in a sentence, search for their weighted sentiment (i.e. some words are more negative,positive and neutral than others) in the lexicon and finally counts the number of words from each sentiment category. In the scoring system, 1 and -1 represent the most positive and the most negative sentiment respectively. For example for a comment labeled as positive in our dataset the scores are 'neg': 0.046, 'neu': 0.781, 'pos': 0.174, 'compound': 0.9609 and for a negative comment in ground truth the scores are as follow 'neg': 0.158, 'neu': 0.755, 'pos': 0.087, 'compound': -0.9011. However, the scores are not always very accurate. For instance another positive comment received the following scores 'neg': 0.044, 'neu': 0.916, 'pos': 0.039, 'compound': -0.1759. We used different forms of VADER scores such as all the raw scores, only raw positive and negative scores and the binary combination of negative and positive scores such that if the subtraction of negative score from positive score was greater than zero the feature was equal to one, otherwise equal to zero.

Finally, due to the massive dimension of the feature matrix, we apply a dimension reduction technique on the final feature matrix in order reduce the number of features. Since the output of the feature matrix is in the form of a sparse matrix, PCA algorithm can not be applied due to its nature (The output will be a dense matrix). We have used the Chi-squared [19] method for this purpose. Chi-squared is a measure of how much dependency exists among the variables; thus if a feature is independent from the target variable, it is an insignificant feature and will be removed from the feature matrix.

4 Approaches

This section focuses on the implementation of Multiple approaches for classification where we test and implement multiple classifiers in order to compare the results and achieve the highest accuracy.

4.1 Bernoulli Naive Bayes

Bernoulli NB classifier is a member of NB classifiers, a family of probabilistic classifiers based on applying Bayes theorem assuming that the features are conditionally independent given the output. In other words:

$$P(x_j|y) = P(x_j|y, x_k),\ \ for\ all\ j, k$$

In the multivariate Bernoulli event model, features are independent Boolean (binary variables) describing the inputs. The model is described as follows:

$$\delta(x) = log\frac{P(y=1|x)}{(y=0|x)} = log(\frac{\theta_1}{1-\theta_1}) + \sum_{j=1}^{m}(x_j log\frac{\theta_{j,1}}{\theta_{j,0}} + (1-x_j)log\frac{1-\theta_{j,1}}{1-\theta_{j,0}})$$

Where θ_1 is the probability of positive output i.e. $P(y = 1)$ and $\theta_{j,1}$ and $\theta_{j,0}$ are probabilities of positive outputs given positive $(P(x = 1|y = 1))$ and negative $(P(x = 1|y = 0))$ outputs respectively. The final output is 1 when $\delta(x) > 0$ and 0 when $\delta(x) < 0$.

4.2 Classification

In this section we discuss multiple classifiers in order to achieve the best accuracy on the final test set. In our experiment we began by using the Multinomial version of the above said naive Bayes classifier in order to account for the non-binary features such as TF*IDF. Different values of smoothing parameter alpha are tested in order to analyze the effect on the classification.

Furthermore, multiple variations of the SVM classifiers were implemented for the comparison purposes. We also implemented the novel NBSVM classifier. We justify Our use of this model based on their highly successful history in classification problems, specially those related to text processing. [20]

In order to get a better sense of different models accuracy, in this specific problem, classifiers such as Decision Tree and K-Nearest Neighbour (KNN) were implemented and compared based on their accuracy. For each classifier, the best hyper parameters were chosen based on experimental results. Finally, two variations of ensemble learning method based on combinations of KNN, Decision Tree, SVM and MNB algorithm using soft voting were implemented to observe the effectiveness of this method for this classification problem.

5 Results

In this section we discuss the results of exploring different proposed models, their parameters and the effect of different features on the performance. We explain the implementation of Bernoulli NB model with binary features and clarify our choice of parameters for our best models.

Training the proposed models using the extracted feature matrices that were occasionally as massive as 25k by 4M was computationally intensive by itself given our available resources. Running cross-validation on each step significantly slowed us down. Consequently, in the early steps of exploring different models and features, we divided the training set to two portions (i.e. 80 % training data and 20 % validation data) to train our model on the training portion and test it on the validation set. At this step we chose our top models that performed the best on the validation portion of the data and developed k-fold cross-validation (k = 4 and 10) on the complete training dataset.

5.1 Bernoulli naive Bayes

Our implemented Bernoulli NB algorithm was tested using a pipeline with BOW (1 to 3) as the features and Chi-squared dimensionality reduction (k = 3M). Using the aforementioned data split, the accuracy on the validation set turned out as 86.84% proving to be functional when compared to the baseline. The confusion matrix and some measures of this algorithm are illustrated in Table 1 and 2.

	actual class (observation)	
predicted class (expectation)	2238 TP	256 FP
	402 FN	2104 TN

Measure	Value
Recall	83.95
Precision	89.15
F1	86.47

Table 1: Confusion matrix of the implemented BNB classifier and several corresponding measures

5.2 Support Vector Machine

In general, SVM appeared to be one of our most successful models. There are four available kernels (i.e. linear, rbf, sigmoid and poly) in the SVM modules of Scikit-Learn that result to different decision boundaries. Among all the kernels, linear kernel gained the minimum of 85 % accuracy (C = 1). Other kernels with different set of parameters (e.g. poly kernel with degree = 3,5,10, gamma = scale, auto) received 50 % and less accuracy. These observations suggested that our data was linearly separable.

Parameter C controls the trade off between misclassification and the margin size by penalizing the misclassified data entries. Increasing C results to narrower margin and increases the accuracy of the classifier on the training set. We examine the effect of different C values (i.e. different values in the range of $1e^{-}3$ to $1e^{5}$). The higher accuracy was gained with C greater than 500. The results just changed in the order of 0.001 (i.e. 0.1 %) with different C values up to $1e^{5}$. to carry on with our experiments we chose linear kernel and set the parameter C to $1e^{3}$.

Different combination of features such as N-gram (N = 1 to 3), TF*IDF, Bing Liu's lexicon[17] and VADER lexicon[18] were examined with our best model. In general the results showed a model with normalized combination of N-gram and TF*IDF features perform similar and better without both binary and frequency version of Bing Liu's lexicon[17]. The former gained the cross-validated accuracy as high as 90.66 % (90.57 % on the test set) while the latter did not reach above 89 % on the validation set. Adding the raw scores of VADER sentiment lexicon[18] did not improve our best accuracy by itself, but the binary feature extracted from the raw scores slightly increased the validation accuracy (91.3 %) while gained slightly worse than our best accuracy on the test set (90.52 %).

The combination of all these features resulted to massive feature matrix (e.g. 25k by 5M) that not all of them are significant predictors of the target. As mentioned in section 4.2 we used Chi-squared module for dimension reduction and selecting our top features. Our best accuracy was gained by selecting top 3M features. It is also worth mentioning that, in almost all of our trials, removing the stop-words resulted to lower accuracy; thus, in our best model we did not remove the stop-words.

As part of the requirements we are asked to try two feature pipelines with one selected model and cross-validation implementation. To do so, we implemented linear SVM (C = $1e^{3}$) to compare N-gram (N= 1,2) with TF*IDF features using 4-fold cross-validation. The cross-validation scores for N-gram features and TF*IDF features are 0.868 (fold scores: 0.869 0.859 0.873 0.871) and 0.808 (fold scores: 0.816, 0.788, 0.811, 0.820) respectively suggesting that TF*IDF is a more powerful feature compare to N-gram.

5.3 Naive Bayes Support Vector Machine

In this section we apply the NBSVM approach proposed in [14] for IMDb sentiment analysis. This method combines generative and discriminative classifiers by building a SVM over NB log-count ratio as feature values. The NBSVM approach performs as the interpolation between NB and SVM prediction. Parameters C = 1000, B = 0.25 were used for the training procedure. A comparison between MNB and

NBSVM classifiers can be seen in the following Table 2. As shown in the Table 2, NBSVM has higher performance compared to MNB in the same circumstance due to the fact that it benefits from both MNB and SVM during its prediction. As mentioned in [14], one disadvantage of NBSVM is having the interpolation parameter and the performance on longer documents is virtually identical for B ∈ [1/4, 1].

Classifier	MNB	NBSVM
Accuracy(%)	86.4	89.2

Table 2: Comparing the performance of MNB and NBSVM

Classifier	Accuracy(Val)	Precision	Recall	F1	Cross Validation Mean
Decision Tree	68.08	45.51	81.95	58.52	68.59
KNearest Neighbour	53.28	**99.19**	51.44	67.75	50.08
SVM	**91.1**	89.89	**91.93**	**90.9**	**87.96**
Multinomial NB	88.67	92.92	82.66	87.49	84.48
Logistic Regression	88.0	86.41	89.0	87.69	86.0

Table 3: Final comparison of the implemented models

5.4 Ensemble Models

Two different sets of ensemble learning method were implemented with the following specifications:

1. **Model 1:** a combination of SVM(C=1000), Decision Tree(max_depth = 4) and KNearest Neighbour(n_neighbour = 6) using soft voting and weight vector of [2,1,1] meaning we put more weight on the outcomes of the SVM classifier. This model resulted in 89.9% accuracy on the validation set.

2. **Model 2:** a combination of SVM(C=1000), Decision Tree(max_depth = 4) , KNearest Neighbour(n_neighbour = 6) and Multinomial naive Bayes(alpha = 6.0) using soft voting and weight vector of [2,1,1,1.5] meaning we put more weight on the outcomes of the SVM classifier and the MnB. This model resulted in 89.92% accuracy on the validation set and 75.81% accuracy on the test set, indicating poor results and a possible meta-overfitting.

References

[1] B. Liu, Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers

[2] Large Movie Review Dataset. (n.d.). Retrieved from: http://ai.stanford.edu/~amaas/data/sentiment/

[3] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

[4] M. Z. Asghar, A. Khan, S. Ahmad, and F. M. Kundi. "A review of feature extraction in sentiment analysis." Journal of Basic and Applied Scientific Research, Vol. 4, no. 3, pp.181-186. 2012.

[5] M. B. Revanasiddappa, B. S. Harish. A New Feature Selection Method based on Intuitionistic Fuzzy Entropy to Categorize Text Documents, International Journal of Interactive Multimedia and Artificial Intelligence, (2018)

[6] A. Sharma and S. Dey. “A comparative study of feature selection and machine learning techniques for sentiment analysis.” In Proceedings of the 2012 ACM research in applied computation symposium, pp. 1-7

[7] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. “Lexicon-based methods for sentiment analysis.” Computational linguistics, Vol. 37, no. 2, pp.267-307. 2011.

[8] A. Mudinas, D. Zhang, and M. Levene. “Combining lexicon and learning based approaches for concept-level sentiment analysis.” In Proceedings of the first international workshop on issues of sentiment discovery and opinion mining, pp. 5. ACM, 2012.

[9] L. Zheng, H. Wang, and S. Gao. “Sentimental feature selection for sentiment analysis of Chinese online reviews.” International journal of machine learning and cybernetics, Vol. 9, no. 1, pp.75-84. 2018.

[10] 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, Association for Computational Linguistics, pp. 79-86. 2002.

[11] A. Tripathy, A. Agrawal, and S.K. Rath. “Classification of sentiment reviews using n-gram machine learning approach.” Expert Systems with Applications, Vol. 57, pp. 117-126. 2016.

[12] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In In Proceedings of the ACL, pages 271–278.

[13] Cui, H., Mittal, V., and Datar, M. Comparative experiments on sentiment classification for online product reviews. In Proceedings of AAAI (Boston, Massachusetts, July 16-20, 2006). 2006, 1265–1270.

[14] Wang, Sida and Manning, Chris D. Baselines and bigrams: Simple, good sentiment and text classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, 2012.

[15] Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. In Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, page e1253. Wiley Online Library.

[16] Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y. and Potts, C., 2011, June. Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1 (pp. 142-150). Association for Computational Linguistics.

[17] Hu and B. Liu. 2004. Mining and summarizing customer reviews. In KDD-2004.

[18] Hutto, C.J. Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[19] Liu, H. and Setiono, R., 1995, November. Chi2: Feature selection and discretization of numeric attributes. In Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence (pp. 388-391). IEEE.

[20] Wang, S. and Manning, C.D., 2012, July. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2 (pp. 90-94). Association for Computational Linguistics.