IMDB Sentiment Analysis

Anne-Marie Zaccarin
260691475

Otman Benchekroun
260663493

Sarah Fenandez
260663523

February 22nd 2019

Abstract

This project evaluates the performance of multiple algorithms in classifying IMDB movie reviews as either positive or negative, and explores the importance of feature construction. The tested models are: Bernoulli Naive Bayes, Multinominal Naive Bayes, Linear Support Vector Machines, Logistic Regression, Random Forests and a One Dimentional Convolutional Neural Network. The impact on accuracy of the number of input features, the use of feature regularization as well as chi squared feature selection is explored. An accuracy of 90,12% on the *Kaggle* subset is achieved using a Linear Support Vector Machine.

1 Introduction

As the use of social media continues to bloom, more and more people voice their opinions online. The abundance of online ratings, recommendations and reviews have large marketing potential. Natural Language Processing (NLP) can be used to analyze this data and establish the reception of a newly launched product, to estimate the popularity of a movie, or to recommend products to a returning customer [1], [2]. Sentiment Analysis, a sub-discipline of NLP, is a vast field pertaining to the classification of opinions in text, whether they be formal as is the case with news articles [3], [4], very short and informal as with tweets [5], [6] or as is the case in this project, audience movie reviews [7],[8], [9], [10].

The goal of this project was to categorize movie reviews as either positive or negative. A database of 50,000 informal International Movie Database (IMDB) movie reviews was used. The performance of multiple models (Bernoulli Naive Bayes (BNB), Multivariate Naive Bayes (MNB), Logistic Regression (LR), Support Vector Machines (SVM), Random Forests (RF) and One Dimensional Convolution Neural Networks (1DCNN)) proven to be effective for sentiment analysis were compared. An accuracy of 90.12% was obtained on the *Kaggle* subset through the use of a Support Vector Machine.

2 Related Work

As sentiment Analysis covers a wide array of problems, many different NLP techniques have been investigated, each method tailored towards its specific application. Since the publication of a large publicly available dataset from IMDB by Maas et al.[7], many researchers have worked on the classification of movie reviews [7],[8], [10].

Maas [7] combines different models to capture both semantic similarities between words and their associated sentiment. A probabilistic model akin to Linear Discriminant Analysis is used to extract similar semantics. LR is employed to map words to their probability of conveying positive sentiments. These combined methods yield a best accuracy of 88.89% on the IMDB dataset when using additional unlabeled data during training.

Mesnil [8] compares the performance of different generative and discriminative models on the same IMDB dataset. A prediction accuracy of 92.57% is achieved by combining and interpolating results from Recurrent Neural Networks (RNNs), Naive Bayes (NB) and SVM models. The RNNs were fed Sentence Vector features (representations of a document by a fixed sized feature vector), while the NB and SVM models used a "bag of words" approach with tri-grams as input features.

Tripathy [10] analyses the effect multiple N-grams inputs have on classification accuracy. Multiple algorithms (NB, Maximum Entropy (ME), SVM, and Stochastic Gradient Descent (SGD)) are tested with various combinations of uni-grams, bi-grams and tri-grams. Uni-grams perform best for ME and SGD with an accuracy of 88,48% and 85,11% respectively. Highest accuracy was obtained by combining uni-grams, bi-grams and tri-grams for NB, with an accuracy of 86,232% and SVM with an accuracy of 88,94%.

3 Dataset and setup

The dataset used in this project contains 50,000 IMDB movie reviews. This data is split into a training set and a testing set, each composed of 25,000 reviews. The training set has an equal number of positive and negative reviews.

Each review in the dataset is provided in an individual *.txt* file. Thus, the first step in pre-processing was to load each review into a list of strings. This list was then shuffled. All characters were lower-cased and each entry tokenized. Furthermore, all stop words (such as *a, and, the, etc.*) were removed from the data set, all using the robust machine learning library of sci-kit learn [11]. Further pre-processing was specific to each algorithm tested and is detailed in the section below.

4 Proposed Approach

We investigated the performance of 5 models whose input parameters follow a bag of words model: BNB, MNB, Linear SVM, LR, RF. Parameter preprocessing for these models exploited the *.CountVectorizer* method from the scikit-learn Python library. We compare the performance of the bag of words based models to a fine-tuned 1DCNN. All accuracy tests performed on these models used 5-fold cross validation. All models were implemented using the scikit-learn library, except for the BNB and the 1DCNN.

NB classifiers have been observed to perform surprisingly well on sentiment analysis, outperforming many other models. They are simple classifiers based on the Bayes probability rule. NB classifiers are very fast to train and are less prone to overfitting than other models [9], [12], [13]. We manually implemented a BNB Classifier and included Laplace smoothing to properly manage words that were not present in the training set. Binary word counts were used as input. In addition, we implemented a MNB classifier and tested it using word counts as input features.

LR is often considered the baseline of supervised learning classification algorithms. When used for Sentiment Analysis, each input parameter's importance in determining the sentiment of a text is established during training [14]. This makes LR very well suited for the bag of words models and has been used with success in establishing sentiment polarity in texts [15], [9].

The arbitrary insensitivity to feature parameters of the RF Model prevents overfitting to the training set when compared to decision tree performance. Its simplicity and low sensitivity to hyperparameter tuning make RFs an interesting option for Sentiment Analysis [16].

Linear SVMs offer an appealing non-parametric approach to classification. They offer a robust option for Sentiment Analysis, as text data is often linearly seperable. Furthermore, the tuning of the compensation factor helps avoid overfitting [17]. Our compensation factor was set to 1.0, this prevented extreme overfitting while conserving a relatively hard SVM boundry.

All these methods have the Bag of Words (BOW) method in common. The BOW model uses a large number of input features. If not restricted, the total dimension of the BOW is the vocabulary size. Furthermore, BOW fail to consider the relation between words. To evaluate the effect of the BOW model on sentiment analysis accuracy, we tested the performance of a 1DCNN. This model

takes embedded sentence vectors as input. We build the 1DCNN ourselves, featuring three 1D Convolution layers (kernels: 7, 5, 3), one Flattening layer, and 2 fully Connected layers (128, 64), with a 0.2 dropout between each layer, built using keras [18]. The Convolutional kernel properties of this NN are what gives it the desired word-order-memorization quality.

Finally, we will evaluate the effects on performance of N-gram features, regularization and feature selection on our linear SVM model.

5 Results

We first compare the performance of all models investigated on basic input parameters. The 1D-CNN was fed text vectors, where each word in the text was replaced by a unique index that represented the popularity of the word through the IMDB corpus. The other models were fed 10,000 BOW uni-gram count vector features. Table 1 describes each model's average accuracy using 5-fold cross validation, save for the neural network, which was evaluated on a separate segment of the data on which it was neither trained nor validated.

Bernoulli NaiveB	MultiVar NaiveB	Random Forest	1DCNN	Logistic Regression	SVM
81.65%	83.21%	81.08%	81.53%	85.12%	85.15%

Table 1: accuracy of all models using described input features

The Linear SVM performs better than any other model. For this reason, we will investigate the effects of different input parameters using this model. We observe that the 1DCNN has one of the lowest accuracy of all models. While at first glance these results suggest that SVMs are the superior model, this does not mean that CNNs are incapable of outperforming SVMs [19]. However, further CNN model design and exploration go beyond the scope of this project.

Figure 1 summarizes the accuracy of linear SVM according to the number of input parameters, regularization methods and feature selection techniques. We first compared the performance of a linear SVM to the number of uni-gram input parameters. A rapid rise in accuracy occurs as the number of words considered increase from 100 to 2,000. When more words are considered, the accuracy drops. We introduce a Term Frequency-Inverse Document Frequency (TF-IDF) regularization parameter and repeat the previous test to determine if the unregularized model was overfitting. With TF-IDF regularization, the more input words are used, the better the accuracy of our model, thus confirming the overfitting theory. We finally try to obtain a better accuracy by selecting our features more efficiently. We use chi squared feature selection available in the scikit-learn library using the *.SelectPercentile()* method. By setting the *percentile* parameter to 40, this method conserves the top 40% most useful input features. When no TF-IDF regularization is applied with the feature selection technique, accuracy peaks at 4,000 input parameters before chi squared selection. When both chi squared feature selection and TF-IDF regularization are applied, the accuracy of linear SVM is maximized at the maximal unigram word count input features value (10,000).

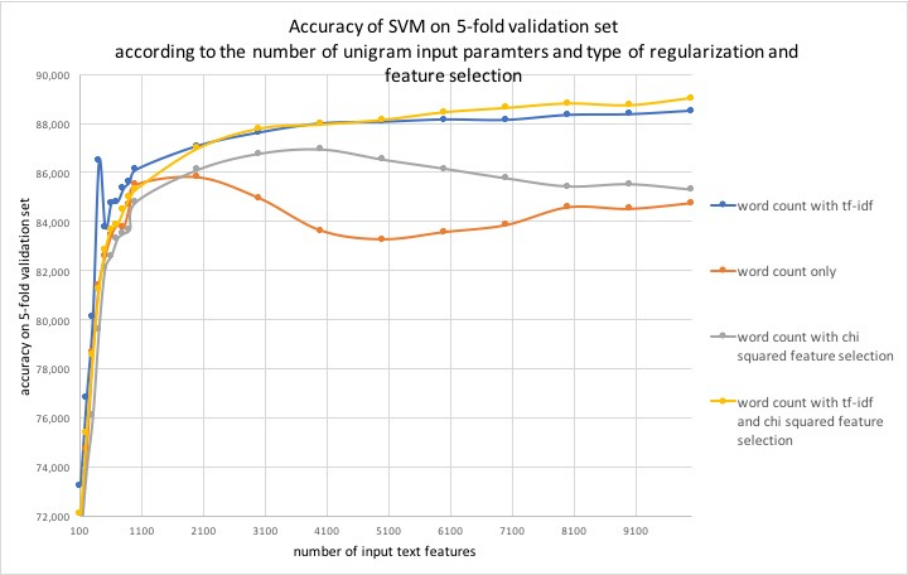

Figure 1: effect of regularization on validation set accuracy

The next test introduces N-grams in the input features in an attempt to counteract the disadvantages of the uni-gram BOW model. This way, the relation between certain words is considered. We observe that the introduction of bi-grams and tri-grams improve the accuracy of the linear SVM during 5-fold cross validation, as detailed in Table 2.

Input Types		Quantity	Regularization	Feature Selection	Accuracy
Unigrams			TFIDF	none	88,57%
	Unigrams	10,000			
Unigrams & Bigrams			TFIDF	none	89,69%
	Unigrams	10,000			
	Bigrams	10,000			
Unigrams, Bigrams & Trigrams			TFIDF	none	89,88%
	Unigrams	10,000			
	Bigrams	10,000			
	Trigrams	10,000			

Table 2: accuracy on 5 fold validation according to input type

Finally, we investigate the effect of chi squared feature selection on our most accurate model, a linear SVM with 10,000 uni-grams, bi-grams and tri-grams as input parameters. The use of chi squared feature selection improves our models performance by only 0,07%, as documented in Table 3. This suggests that the impact of feature selection decreases when very large amounts of input parameters are used.

Input Types		Quantity	Regularization	Feature Selection	Accuracy
Uni-grams, Bi-grams & Tri-grams			TF-IDF	none	89,87%
	Uni-grams	10,000			
	Bi-grams	10,000			
	Tri-grams	10,000			
Uni-grams, Bi-grams & Tri-grams			TF-IDF	chi^2	89,94%
	Uni-grams	10,000			
	Bi-grams	10,000			
	Tri-grams	10,000			

Table 3: effect of chi squared feature selection on accuracy for 5 fold cross validation

6 Discussion and Conclusion

Linear SVMs are a strong candidate model for Sentiment Analysis tasks. While a simple uni-gram BOW model leads to acceptable prediction accuracy, the inclusion of bi-grams and tri-grams improves the models performance. The importance of feature selection is outlined by the accuracy improvements brought on by the use of chi-squared feature selection, set at 40%. Finally when using large quantities of input features, regularization, here TF-IDF, is essential to limit overfitting. While linear SVM proved efficient at Sentiment Analysis, other models also performed quite well. Ensembling of many different models, such as SVMs and CNNs might offer noteworthy accuracy gains.

7 Statement of Contributions

Anne-Marie was responsible for implementing the Bernoulli Naive Bayes as well as testing the performance of the Support Vector Machine for different input parameters. Otman implemented the data pre-processing, 1D Convolution Neural Network, and the k-fold cross validation splitting. Sarah was responsible for the logistic regression and decision tree algorithms. All team members contributed equally to the report.

References

[1] Alex Wright. Mining the web for feelings, not facts. *New York Times*, Aug 2009.

[2] Meena Rambocas, João Gama, et al. Marketing research: The role of sentiment analysis. Technical report, Universidade do Porto, Faculdade de Economia do Porto, 2013.

[3] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.

[4] Robert P Schumaker, Yulei Zhang, Chun-Neng Huang, and Hsinchun Chen. Evaluating sentiment in financial news articles. *Decision Support Systems*, 53(3):458–464, 2012.

[5] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[6] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! In *Fifth International AAAI conference on weblogs and social media*, 2011.

[7] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher 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*, pages 142–150. Association for Computational Linguistics, 2011.

[8] Grégoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato, and Yoshua Bengio. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *arXiv preprint arXiv:1412.5335*, 2014.

[9] Sida Wang and Christopher D Manning. 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*, pages 90–94. Association for Computational Linguistics, 2012.

[10] Abinash Tripathy, Ankit Agrawal, and Santanu Kumar Rath. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, 57:117–126, 2016.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[13] Vivek Narayanan, Ishan Arora, and Arjun Bhatia. Fast and accurate sentiment classification using an enhanced naive bayes model. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 194–201. Springer, 2013.

[14] Dan Jurafsky and James H Martin. *Speech and language processing*, volume 3. Pearson London, 2014.

[15] Hussam Hamdan, Patrice Bellot, and Frederic Bechet. Lsislif: Crf and logistic regression for opinion target extraction and sentiment polarity analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 753–758, 2015.

[16] Xing Fang and Justin Zhan. Sentiment analysis using product review data. *Journal of Big Data*, 2(1):5, 2015.

[17] Laura Auria and Rouslan A Moro. Support vector machines (svm) as a technique for solvency analysis. 2008.

[18] François Chollet et al. Keras. https://keras.io, 2015.

[19] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.