Sentiment Analysis And Tag Recommendation For Online MultiMedia

Afshin Moin (afshinm), Abhishek Bharani (abharani), and Peng Seng Kuok (pkuok)

1 Introduction

Content sharing has been the key ingredient of the biggest success stories in the last decade. Social networks like Facebook and LinkedIn, video sharing services like YouTube, Ecommerce websites like Amazon and online travel agencies like TripAdviser and Expedia are all examples of online platforms that provide a convenient way for their users to exchange content, goods or services. It is obvious that the success of such online platforms depends on how effectively they link the end users to relevant content. This latter cannot be achieved without proper content classification and user feedback analysis. This motivated us to apply machine learning techniques to automate the process of user feedback analysis and content classification. We work with a real sample dataset taken from YouTube [3]. Nevertheless, same techniques can be applied to any other database with similar structure like Yelp and Netflix. In Section 3, we discuss our approach to data labeling and feature extraction. In Section 4, we review our approach to sentiment and category classification of YouTube comments. Results of the experiments for different classification techniques are compared with each other in Section 5. Also, calibration curves are presented and effect of dimensionality reduction on accuracy is examined. Section 6 concludes the report.

2 Related Work

3 Dataset And Feature Extraction

In this section we discuss the properties of the dataset we use for our experiments as well as feature extraction techniques adopted.

3.1 YouTube Dataset

We apply classification techniques on a sample dataset taken from YouTube [3]. IThis dataset includes 691407 comments for 200 most trending YouTube videos in US in a two weeks period. The same source also offers a similar dataset for the most trending videos in GB. We will use a fraction of this dataset as our test set for sentiment analysis. The data includes comment data and video data. The comment data maps video IDs to comments and the number of likes and dislikes for the corresponding comments. The video data contains video title, channel title, category, tags, number of views, likes and dislikes and user reviews. The US videos belong to 15 categories.

3.2 Label Generation

Our goal is to classify YouTube comments into positive, negative or neutral sentiment classes. However, there are no labels showing the sentiment of the comments. Considering the large number of comments, it was not possible for us to manually label a reasonable fraction of this dataset. Consequently, we used the TextBlob [2] library for Python to generate the labels. TextBlob applies NLP (Natural Language Processing) techniques to

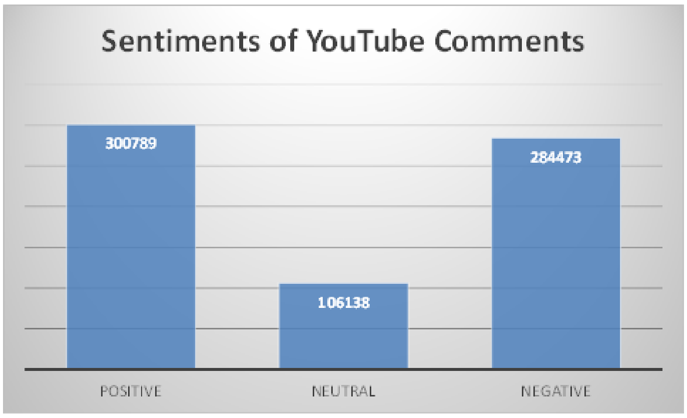

Fig. 1. YouTube Dataset Statistics

estimate the **polarity** of each comment independently from the rest of the comments, and only based on its content. For us, it replaces the burdensome task of manual labeling. TextBlob generates polarity scores in the range of $(-1, 1)$. We convert them into categorical data using Equation 1. Statistics on the size of class is shown in Figure 1.

$$\text{sentiment} = \begin{cases} -1 & \text{polarity} < 0 \\ 0 & \text{polarity} = 0 \\ 1 & \text{polarity} > 0 \end{cases} \tag{1}$$

Words that happen more frequently in positive and negative comments are shown as word clouds in Figures 2 and 3.

In Section 5.3, we do category classification based on tags and comments. The category labels are already available in video data.

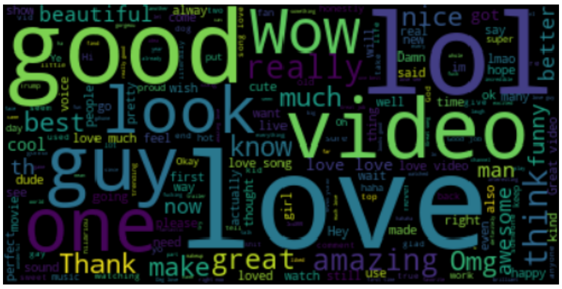

Fig. 2. Word cloud for positive comments.

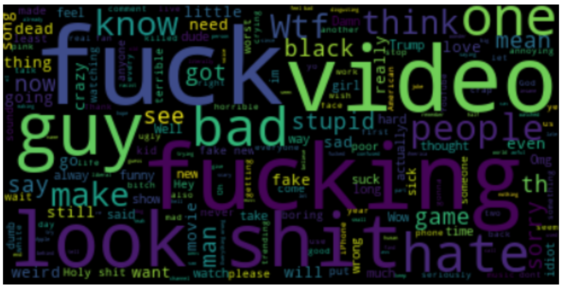

Fig. 3. Word cloud for negative comments.

3.3 Feature Extraction

For sentiment classification based on comments, we used TF-IDF [1] to convert comments into feature vectors. In this technique, TF (Term Frequency) counts the number of times a word has occurred in each document. However, the number of occurrences of words is in general higher for longer documents. Then, it is helpful to divide the number of word occurrences by the number of words in the document. The output of TF is a sparse matrix mapping documents to a normalized vector representing how many times each word has occurred in them. IDF (Inverse Document Frequency) accounts for the number of occurrences of a word in the entire document corpus. Namely, some words are more likely than others to happen in a given corpus of documents. Then, we scale TF terms by a decreasing function of the number of occurrences of each word in the whole corpus which is indeed the IDF term.

For category classification, we took two approaches. First, we used the same TF-IDF comment features. Second, we used the tags from the video data. Since video data has been gathered daily over two weeks, it may contain each video multiple times along with different tags. To remove duplicate entries, we removed punctuations and English stop words from the tags. Stop words are first removed. Stop words are commonly used words such as *the* which are filtered out before the analysis. Then, the videos were grouped based on their ID and tags were aggregated taking their union. We denote the tags of a video by a binary feature vector where each entry is zero in case the video was tagged by the corresponding tag and zero otherwise.

4 Content Classification Approach

We used 4 supervised learning methods for the problems of sentiment and category classification. In both cases we deal with multi-class classification problems. In sentiment analysis, there are 3 classes corresponding to negative, neutral and positive sentiments while in category classification there are 15 classes corresponding to different categories. Theses techniques include Logistic Regression, Ridge classifier, Bernoulli Naive Bayes classifier and linear Support Vector Machine classifier.

Bernoulli Nave Bayes model is considered a common baseline for text classification due to its speed. This model assumes conditional independence between the features given their class. This assumption leads to relatively straightforward analysis and efficient running time. Nevertheless, the performance is less desirable than other methods because the independence condition is usually not satisfied in real-world problems.

Multinomial Logistic Regression, also known as softmax regression is a generalization of Logistic Regression to more than 2 classes. Matrix of parameters θ is found maximizing the log-likelihood function defined as:

$$TobeFilled \tag{2}$$

The probability of each data point is then computed as:

$$TobeFilled \tag{3}$$

Ridge classification is a generalization of regularized linear regression to classification problems. The cost function to minimize if least squares error with L2 norm defined as:

$$Tobefilled \tag{4}$$

Once parameters are computed, continues predicted values are done according to a linear prediction function $X\theta$. These values are then converted into discrete class values through proper thresholds.

The second term is used to penalize the matrix W being too large. In other words, if the matrix W takes on large values, regularized loss function will be penalized. This will encourage the fitted model to be a simple model rather than a complex model and usually this will prevent overfitting.

Linear Support Vector classifier (LinearSVC) applies a Support Vector Machine (SVM) with a linear Kernel. SVM is a maximum-margin classifier seeking to find a hyperplane

$W^T X + b$ in a high-dimensional space that solves the following constrained optimization problem.

$$Tobefilled \tag{5}$$

Support Vector Machines are inherently binary classifiers. To generalize them to the case of multi-class problems, there are two possible methods: one-vs-one and one-vs-all. The one-vs-one method builds a classifier for every two classes. The class of a point is then the one chosen by the most classifiers. The one-vs-all approach builds a classifier for each class compared to all the remaining classes. The class of a data point is the one whose classifier achieves the greatest margin. In our experiment, we apply the one-vs-all method.

5 Experiments and Evaluation

In this section, we present our experimental methodology, evaluation metrics and the results of the experiments.

5.1 Sentiment Analysis

In order to measure the performance of different classification techniques on the YouTube data, we split the data from US comments into train and test subsets. To generate the train set, we randomly choose 80% of the comments from each of the three categories. The remaining comments form the test set. We use 20% of the GB comments as development set. Table 1 shows the accuracy measured on train and test sets. Naive Bayes classifier has the worst performance among all methods. This result is predictable as the conditional independence condition required by the Naive Bayes classifier is not satisfied in this dataset. In particular, the probability of two words happening in a comment given the category of the comment are not independent from one another. On this dataset, SVM classifier has the best performance.

Model	Dev Accuracy	Test Accuracy
Logistic Regression	0.709	0.709
Bernoulli NB	0.709	0.715
Ridge Classifier	0.914	0.905
Linear SVC	0.956	0.951

Table 1. Sentiment classification accuracy of YouTube comments

Figure 4 shows the calibration curve of the aforementioned techniques. It is observed that Naive Bayes classifier is not well-calibrated over the range $[0, 1]$. From among the remaining three methods, Logistic Regression is well-calibrated over the range $[0, 1]$ and is in general better calibrated than the other two methods.

5.2 Effect of Dimensionality Reduction

In this set of experiments, we examined how dimensionality reduction can affect the accuracy of predictions. Dimension reduction is usually applied to reduce the complexity of computations. We apply TruncatedSVD to project TF-IDF features to fewer dimensions than those of the original TF-IDF feature space. We use TruncatedSVD rather than Principal Component Analysis (PCS) because the TF-IDF features are stored in a sparse matrix by the SKLearn library which PCA cannot process. TruncatedSVD is very similar

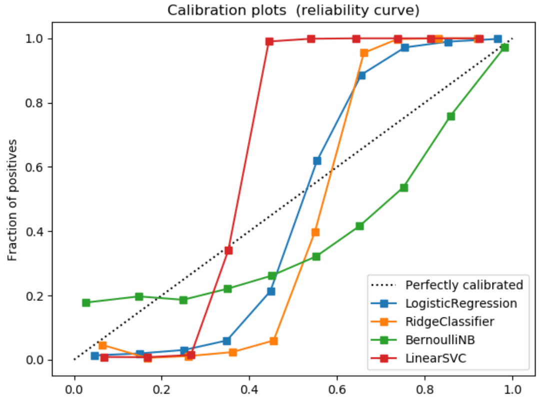

Fig. 4. Word cloud for positive comments

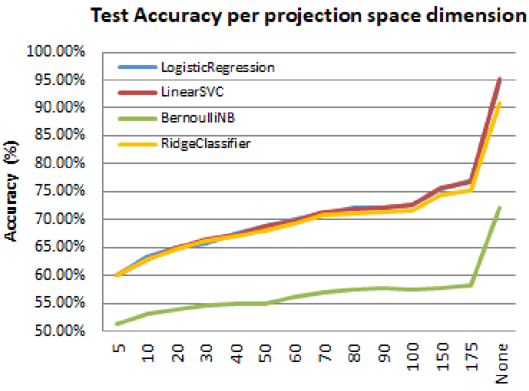

Fig. 5. Accuracy vs. project space dimensions

to PCA. However, the data is not centered around its mean in the former. Figure **??** shows the accuracy of sentiment analysis with TruncatedSVD applied to TF-IDF features versus number of dimensions of the projection space. It is observed that accuracy increases with the number of dimensions. Nevertheless, with 175 dimensions, it is still far from the case where no dimensionality reduction is applied. It seems that in this dataset, 175 dimensions is not enough to project the data without loosing much information. We could not experiment with higher dimensions due to memory limitations of our desktops.

5.3 Category Classification

First, we report the results of experiments for category classification using TF-IDF comment features. In this method we try to detect the category of the video a comment is talking about using the TF-IDF features of the comments. Table **??** shows the accuracy of different techniques. It is seen that the results are not as good as sentiment classification. Considering there are 15 categories versus 3 sentiment classes, this problem is deemed to be harder. Yet, the probability of a successful random category guess is only 1/15(6.7%). Consequently, category prediction based on TF-IDF features of comments is still a big improvement over random guessing. We then conducted category classification based on video tags. Table **??** shows accuracy of category classification using tags. It is observed that tags can predict the category with much higher accuracy. Note that tags are considered a property of videos while TF-IDF features are a property of comments. In addition, tags are explicitly chosen to classify the videos. As a result, it is expected that are more directly relevant to the content of the video and are less noisy which lead to better accuracy in category prediction.

In another experiment, we examined the effect of the number of categories on the accuracy. Intuitively, the larger the number of categories, the more difficult the classification problem is expected to be. To experiment this intuition, we created a list of categories with ascending order of each category population. We considered the first 2 categories, that is, the two categories with greatest number of videos, ignored the remaining videos in train and test set and measured the accuracy. The same experiment was repeated for $3, 4, \ldots, 15$ categories. Figure **??** shows how test accuracy decreases by increasing the number of categories.

6 Conclusion

References

1. A. Rajaraman and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2011.
2. YouTube. TextBlob Library For Python. https://textblob.readthedocs.io/en/dev.
3. YouTube. YouTube Database For Kaggle Competitions. https://www.kaggle.com/datasnaek/youtube.

7 Contributions

All of us contributed in the discussions about what problem to target, what dataset to use and what techniques to apply. All of use helped with writing and reviewing the report. Peng searched different potential dataset candidates and summarized their advantages and shortcomings. He generated statistics about the YouTube dataset. Abhishek did sentiment polarity analysis and generated word clouds using TextBlob. Afshin implemented feature extraction and conducted classification experiments using scikit.