Mini Project 2: Classification of Textual Data

Group 65
Alhusain Abdalla, Rossen Vladimirov, Ege Yay

Abstract

In this project, we used supervised learning for classifying textual data in two datasets. The first dataset contains posts in 20 newsgroups, while the second dataset consists of IMDB reviews. We conducted multi-class classification on the 20 newsgroups dataset and predicted the newsgroup to which a post belongs. We also conducted binary classification on the IMDB reviews dataset to predict whether a review is positive or negative. We built a data pre-processing pipeline to perform tokenization and stemming as well as exclude stopwords and tokens with non-alphabet characters. From each newsgroup post and IMDB review, we extracted six features: word frequencies, emotion word frequencies, sentence emotion score, part-of-speech frequencies, post/review length, and sentence length. Following that, we built a validation pipeline which uses k-fold cross-validation on the train set to evaluate the effect of different hyper-parameter values and features. After evaluating the final accuracy of all models on the test set, the best-performing model on the 20 newsgroups dataset is support vector machine (SVM) with 69.20% accuracy, while the best-performing model on the IMDB reviews dataset is logistic regression with 88.32% accuracy. The combined accuracy on both datasets is 83.89%.

1 Introduction

In machine learning, multi-class classification is used to classify instances into categories (classes). To aid this, sentiment analysis is a useful tool for interpreting and classifying emotions (positive or negative) within text data using different machine learning and natural language processing techniques for text analysis.

In this project, we used supervised learning for classification of textual data, where we worked with two datasets: 20 newsgroups dataset and IMDB reviews dataset.

We conducted multi-class classification on the 20 newsgroups dataset and built a model which predicts the newsgroup to which a post belongs based on the post's text.

Similarly, we conducted binary classification on the IMDB reviews dataset to predict whether a movie review is positive or negative based on the review's text.

First, we pre-processed the post and review text by performing tokenization, stemming with NLTK's Porter stemmer, and elimination of stopwords as well as tokens containing non-alphabet characters.

Next, from each newsgroup post and IMDB movie review, we extracted six features: word frequencies, emotion word frequencies, sentence emotion score, part-of-speech frequencies, post/review length, and sentence length.

In addition, we built a validation pipeline with k-fold cross-validation on the train set to evaluate the effect of different hyper-parameter values and features. For each dataset and classifier, we picked the best model by selecting features and hyper-parameter values with the highest mean accuracy.

Finally, we re-trained the best model (feature and classifier with fine-tuned hyper-parameters) on the full train set and evaluated the final accuracy on the test set. All models performed satisfactorily, where the average accuracy across all models was in the low 60s for the 20 newsgroups dataset and in the mid 80s for the IMDB reviews dataset.

2 Related Work

2.1 Sentiment Analysis

Maas et al. (2011) developed a model which uses supervised and unsupervised methods to learn word vectors that capture semantic information as well as sentiment. This model leverages both continuous and multi-dimensional sentiment information as well as non-sentiment annotations. The authors tested the model on the IMDB reviews dataset which we use in our experiments.

2.2 Reviews Studies

Hong et al. (2017) performed a meta-analysis of 42 studies on online review helpfulness. They categorized the main review features into several groups: review depth, review readability, linear and quadratic review ratings, and review age.

They found that review depth and review age had a positive influence on perceived helpfulness, while there was no statistically significant relationship between review readability and review rating features and helpfulness.

2.3 Emotionality

Martin and Pu (2014) examined the relationship between emotionality of reviews and helpfulness by comparing models trained with different types of features on three review datasets (TripAdvisor, Yelp, and Amazon).

The authors looked at features such as review depth, review readability, and product rating, as well as TF-IDF scores and part-of-speech tags. To measure emotionality, they used a lexicon of words associated with emotions and counted the number of words in the review text belonging to the lexicon.

They analyzed the results from sentiment analysis of the reviews. Emotionality features performed well for classifying reviews as helpful or unhelpful, but the part-of-speech tag feature set had the best performance on the Amazon review dataset.

2.4 Linguistic Categories

Krishnamoorthy (2015) investigated the relationship between the use of linguistic category features and perceived helpfulness of reviews by training several types of classification models on Amazon review data.

The categories of interest were adjectives, state verbs, state action verbs, interpretive action verbs, and descriptive action words. The author classified words with a model built on top of NLTK's part-of-speech tagger and computed normalized occurrence counts of the categories for each review.

The author also looked at product rating, review age, various readability scores, and subjectivity features (positive and negative opinion words). Models using linguistic category features performed better than models using readability scores or subjectivity features, but the best performance came from models trained on a combination of feature sets.

3 Datasets and Setup

3.1 Datasets

The 20 newsgroups dataset contains 18,000 newsgroups posts on 20 topics split in two subsets: one for training and one for testing. The split between the train and test set is based upon messages posted before and after a specific date.

The IMDB reviews dataset contains movie reviews and their associated sentiment polarity labels. The core dataset has 50,000 reviews split into 25,000 train and 25,000 test sets. In the collection, there are no more than 30 reviews per movie because reviews for the same movie tend to have correlated ratings.

3.2 Data Pre-Processing

Each dataset was pre-processed according to the following procedure. First, we split the text of each newsgroup post and IMDB review into tokens using NLTK's word_tokenize() function.

Next, we removed all punctuation characters as per the list in Python's string class. In addition, we used string's isalpha() method to exclude from consideration all tokens containing non-alphabet characters. We then converted all remaining tokens to lowercase with string's lower() method.

During the next pre-processing step, we excluded all stopwords from the text. The list of stopwords in the English language comes from the NLTK corpus. During the last pre-processing step, each token was stemmed using the Porter stemmer from the NLTK library.

After pre-processing, each newsgroup post and IMDB review is represented as a list of tokens in lowercase, where each token is a stemmed word and none of the tokens is a stopword in the English language.

4 Proposed Approach

4.1 Feature Extraction

We extracted six features (word frequencies, emotion word frequencies, sentence emotion score, part-of-speech frequencies, post/review length, sentence length) from each newsgroup post and IMDB review. Each feature captures different information about posts and reviews and belongs to one of the following categories.

4.1.1 Word Frequencies Feature

We used scikit-learn's TfidfVectorizer to build a feature vector with the frequencies of the stemmed words excluding stopwords and tokens with non-alphabet characters.

Naturally, we expect to observe a relationship between the occurrence of certain words in the text of the post and the newsgroup to which the post belongs. We also speculate that some words appear more often in positive reviews and other words appear more often in negative ones.

4.1.2 Emotionality Features

We used the VADER (Valence Aware Dictionary for sEntiment Reasoning) lexicon of emotion words provided by NLTK. Sentiment analysis based on VADER is sensitive to the polarity (positive vs. negative) as well as the intensity (strength) of the emotion.

For the emotion word frequencies feature, we calculated the proportion of words in a newsgroup post and IMDB review that occur in the VADER lexicon of emotion words. For the sentence emotion score feature, we built a feature vector by using VADER to determine the compound sentiment score of each sentence in a newsgroup post and IMDB review.

We expect that reviews expressing strong emotions might be more positive or negative. For example, emotion words like 'love', 'joy', 'laugh', 'happy', 'optimistic', 'satisfied' are often found in positive reviews, while emotion words like 'angry', 'sad', 'furious', 'fear', 'regret' would appear more often in negative reviews.

4.1.3 Part-of-Speech Feature

We used NLTK's built-in part-of-speech tagger to calculate the frequency with which each tag from the Penn Treebank tagset occurs in the newsgroup post or IMDB review text. We speculate that using certain parts of speech contributes to a review being positive or negative.

4.1.4 Structural Features

Structural features include newsgroup post and IMDB review length in number of characters as well as sentence length. Hong et al. (2017) found review length to be related to review helpfulness since longer reviews are more informative.

We investigated if there is a relationship between the length of a review and it being positive or negative. We also investigated the relationship between the length of a post and the newsgroup to which it belongs. We speculate that, in general, posts in certain newsgroups are longer or shorter than posts in other newsgroups.

4.2 Models

Following our feature extraction phase, we applied and compared the performance of five classifiers implemented by the scikit-learn library: logistic regression, decision tree, support vector machine, AdaBoost, and random forest.

4.2.1 Logistic Regression

Logistic regression is a classification algorithm that takes a linear combination of the input and transforms it using the sigmoid function (for binary classification) or softmax function (for multi-class classification). The output is a probability which is mapped to one of several discrete classes.

4.2.2 Decision Tree

Decision tree is a supervised learning algorithm that creates a tree for predicting the value of a target variable (e.g., class label), where each node splits the training set into several regions based on the value of one feature. Each leaf represents a region of the input space in which examples are assigned the same label. Although decision trees are not very sensitive to outliers, they can easily overfit the training data.

4.2.3 Support Vector Machine

Support vector machine (SVM) is a machine learning algorithm used for regression and classification. It considers each data item as a point in n-dimensional space, where n is the number of features. The algorithm performs classification by finding the maximum-margin hyper-plane that separates the classes.

4.2.4 AdaBoost

AdaBoost is a machine learning meta-algorithm used together with other learning algorithms to improve their performance. The output of weak learners is combined into a weighted sum representing the output of the boosted classifier. The algorithm is adaptive as subsequent weak learners are tweaked in favor of instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers.

4.2.5 Random Forest

Random forest is a classification algorithm which consists of many decisions trees built using bagging and random feature selection. Its main goal is to create an uncorrelated forest of trees whose prediction is more accurate than the predictions of the individual trees.

4.3 K-Fold Cross-Validation (CV)

We developed a model validation pipeline with k-fold cross-validation to evaluate the effect of different hyper-parameter values and design choices (features).

We used scikit-learn's GridSearchCV class to perform k-fold cross-validation on the train set for each dataset (newsgroup posts and IMDB reviews), classifier and feature combination to tune the relevant hyper-parameters. The train set was split into k folds. Each time k−1 folds were used as a training set and 1 fold was used as a validation set.

In our tests, we used 5-fold cross-validation. For each classifier, we picked the feature and hyper-parameter values corresponding to the best mean accuracy. We then re-trained the best model (feature and classifier with fine-tuned hyper-parameters) on the full train set and calculated the final accuracy by calling the classifier's predict() method on the test set.

4.4 Hyper-parameters

The following table describes the relevant hyper-parameters for different classifiers and their corresponding values used in the 5-fold cross-validation in our experiments.

Classifier	**Hyper-parameters**
Logistic Regression	solver: ['lbfgs', 'newton-cg'] C: [1.0, 0.1] tol: [0.01, 0.001]
Decision Tree	criterion: ['gini', 'entropy'], splitter: ['best', 'random'], min_samples_leaf: [1, 5, 10]
SVM	C: [1.0, 0.1, 0.01, 0.001]
AdaBoost	base_estimator: [Perceptron(), SGDClassifier()]
Random Forest	criterion: ['gini', 'entropy'], max_features: ['sqrt', 'log2'], min_samples_leaf: [1, 5, 10]

5 Results

For each dataset, the best feature and hyper-parameter values are selected using 5-fold cross-validation on the train set. The final accuracy of each model is calculated on the test set.

5.1 20 Newsgroups Dataset

Classif.	**Best feature**	**Best hyper-parameter values**	**Mean accuracy train set 5-fold CV**	**Accuracy test set**
Log. Reg.	word freq.	solver: lbfgs C: 1.0 tol: 0.01	72.30	67.37
Decision Tree	word freq.	criterion: gini split: random min_sa_leaf: 1	44.87	41.12
SVM	word freq.	C: 1.0	76.01	**69.20**
Ada Boost	word freq.	base_estim: Perceptron	69.82	63.08
Random Forest	word freq.	criterion: gini max_feat: log2 min_sa_leaf: 1	64.07	60.38

The best performing model on the 20 newsgroups dataset is support vector machine (SVM) with the word frequencies feature achieving 69.20% accuracy on the test set.

5.2 IMDB Reviews Dataset

Classif.	Best feature	Best hyper-parameter values	Mean accuracy train set 5-fold CV	Accuracy test set
Log. Reg.	word freq.	solver: lbfgs C: 1.0 tol: 0.01	88.70	**88.32**
Decision Tree	emot. word freq.	criter: gini split: random min_s_leaf: 10	74.70	75.19
SVM	word freq.	C: 0.1	89.34	87.72
Ada Boost	word freq.	base_estim: SGDClassifier	86.07	83.66
Random Forest	word freq.	criter: entropy max_feat: sqrt min_s_leaf: 5	84.27	84.42

The best performing model on the IMDB reviews dataset is logistic regression with the word frequencies feature achieving 88.32% accuracy on the test set.

5.3 Model Performance Comparison on Both Datasets

Classifier	Accuracy on the test set (%)	
	20 newsgroups	IMDB reviews
Log. Reg.	67.37	**88.32**
Decision Tree	41.12	75.19
SVM	**69.20**	87.72
AdaBoost	63.08	83.66
Random Forest	60.38	84.42

The combined accuracy on both datasets is 83.89%.

After implementing binary classification on the IMDB reviews dataset and multi-class classification on the 20 newsgroups dataset, we observe that all models perform better on the IMDB reviews dataset than on the 20 newsgroups dataset. These results confirm our expectations that, in general, binary classification is an easier problem to solve than multi-class classification.

6 Discussion and Conclusions

We used supervised learning to classify textual data. We conducted multi-class classification on the 20 newsgroups dataset to predict the newsgroup to which a post belongs, as well as binary classification on the IMDB reviews dataset to predict if a review is positive or negative.

We created a data pre-processing pipeline to perform tokenization, stemming, and exclusion of stopwords and non-alphabet tokens. From each post and review, we extracted six features: word frequencies, emotion word frequencies, sentence emotion score, part-of-speech frequencies, post/review length, and sentence length.

In addition, we built a validation pipeline with k-fold cross-validation on the train set to tune different hyper-parameter values and pick the best features. We re-trained the model on the full train set and calculated the accuracy on the test set. The best performing model on the 20 newsgroups dataset is support vector machine (SVM) achieving 69.20% accuracy, while the best performing model on the IMDB reviews dataset is logistic regression achieving 88.32% accuracy.

Our implementation demonstrates the importance of data pre-processing as well as classifier and feature selection for achieving high accuracy. For both datasets, the best feature was found to be word frequencies. This is because posts in the same newsgroup or reviews with the same polarity use common terms. For the IMDB reviews dataset, the emotion words feature ranked a close second for the same reason.

An area for future investigation would be adding more hyper-parameters for k-fold cross-validation as well as expanding the set of values for the existing ones. Increasing the number of hyper-parameters and values to tune would increase the running time of training.

Another area for investigation is combining two or more features in one feature vector in order to find the best model. For example, an emotionality feature can be combined with a structural one.

Lastly, we can experiment with more classifiers beyond the list of six shown here.

7 Statement of Contributions

Rossen Vladimirov implemented the data pre-processing, feature extraction and model validation pipeline with k-fold cross-validation. Alhusain Abdalla and Ege Yay experimented with hyper-parameter tuning.

References

Steven Bird, Ewan Klein, and Edward Loper (2009), Natural Language Processing with Python, O'Reilly Media.

Hong Hong, Di Xu, G. Alan Wang, Weiguo Fan. 2017. Understanding the determinants of online review helpfulness: A meta-analytic investigation. Decision Support Systems, 102:1-11.

C.J. Hutto, E.E. Gilbert, (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.

Srikumar Krishnamoorthy. 2015. Linguistic features for review helpfulness prediction. Expert Systems with Applications, 42(7):3751-3759.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2011), pages 142–150, Portland, USA.

Lionel Martin and Pearl Pu. 2014. Prediction of helpful reviews using emotions extraction. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI). AAAI Press, pages 1551–1557.

Pedregosa et al., Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.