News Articles Classification: Bag-of-Words approach

Kushal Kokje and Udit Patel
kkokje@iu.edu udipatel@iu.edu
December 7,2016

1. Introduction

The goal of this project is to develop a model that will classify news articles based on topics like Business, Tech, Sports, Politics and Entertainment. The classifier will be built using Bag-of-words model. The Document-Term matrix, a high dimensional sparse matrix is built from the raw text files of the news articles. The Raw text files will be transformed into a sparse Matrix using advanced Natural Language Techniques like Vectorization, tf-idf transformation and removing "English" stop words. The focus is on choosing the best classifier from a variety of Machine learning algorithms and using cross validation and optimization techniques for best results.

1.1 Data Collection and Data set

For this task we collected news articles from BBC news dataset [1]. It consists of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005.Natural Classes: 5 (business, entertainment, politics, sport, tech). The Dataset was split into 80%-20 % for training set and testing set respectively. The full dataset was used for cross validation technique. In order to avoid the skewed data, special python packages were used to make sure both the test and train sets have balanced class distribution across all documents. Class label Distribution for the train set y_train is {**'business'**: 408, **'entertainment'**: 302, **'politics'** :339, **'sport'**: 408, **'tech'** :323} and for the test set y_test is {**'business'**: 102, **'entertainment'**: 84, **'politics'** :78, **'sport'**: 103, **'tech'** :78}.

2. Method

2.1 Feature Extraction

In order to perform machine learning we need to convert the raw news texts to feature vectors. We used bag of words model for news classification. Feature set was based on a dictionary of words extracted from the raw news texts data. Features represent individual word fragments (tokens) along with numeric features that exists in raw text files. The features set excludes "English" language stop words from the NLTK standard corpus. To downscale the weights for the words that

appears many times in the document (and hence are less informative), we used the TF-IDF transformation (Term Frequency times Inverse Document Frequency). The final feature set consists of 29421 distinct tokens from 2225 total news articles. Figure 1 demonstrates the end to end classification process.

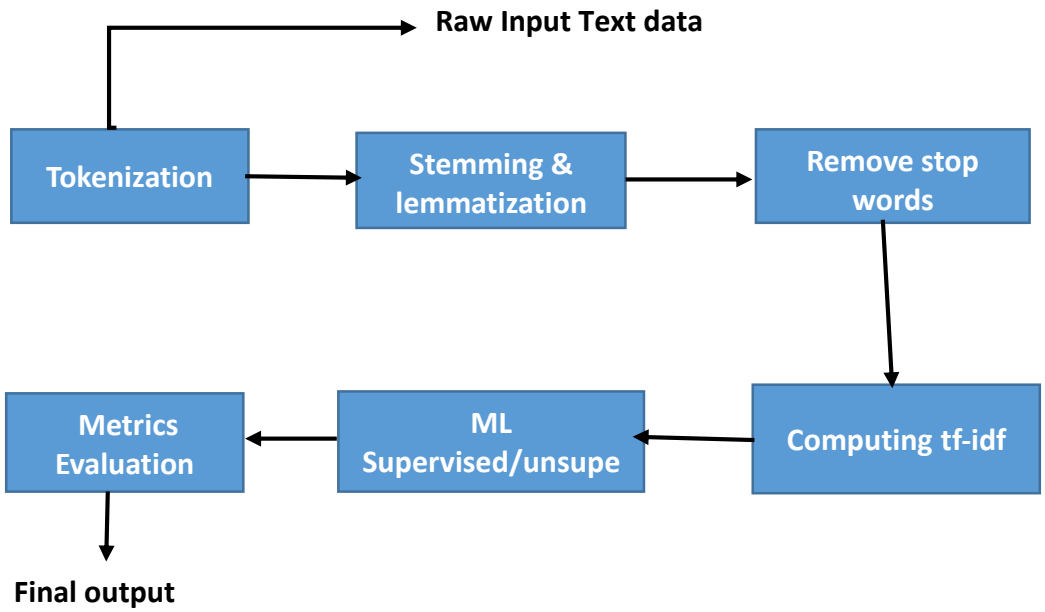

Figure 1. Flow diagram for the end to end classification process

2.1 News articles classification

For news article classification, our approach was to build a base model by testing a set of classifiers and checking the accuracy. The classifiers used were Decision Trees, Logistic Regression, Naïve Bayes, Support Vector Machine, Ensemble methods like Radom Forrest and Adaboost and Gradient boosting. We also used K-means clustering to cluster the unlabeled data into 5 categories. For all the methods 80% of the data was used for training and rest 20% for testing. For unsupervised approach only 30% of the total data with equal class distribution was used for better visualization and understanding. The results of the classifiers can be further found in section 3: Results. We further exploit Naive Bayes and Support Vector Machines as both the algorithms are ideal for Text Classification process.

2.2 Naïve Bayes

We used MultiNomial Naïve Bayes which is a probabilistic learning method. The parameter value alpha was chosen as 0.5 after running a grid search with various alpha values and testing the accuracy on the test set. The accuracy with alpha = 1 we got is 84.26%. The classification report with the various performance metric is as shown in Figure 2

Classification report [Naïve Bayes with alpha = 1]

Class	precision	recall	f1-score	support
business	0.77	0.98	0.86	102
entertainment	0.97	0.44	0.61	84
politics	0.99	0.86	0.92	78
sport	0.74	1.00	0.85	103
tech	0.97	0.87	0.92	78
avg / total	0.87	0.84	0.83	445

Confusion Matrix

business	[100, 0, 0, 1, 1]
entertainment	[17, 37, 0, 29, 1]
politics	[7, 0, 67, 4, 0]
sport	[0, 0, 0, 103, 0]
tech	[6, 1, 1, 2, 68]

Figure 2: Classification report and Confusion matrix for Naïve Bayes with alpha = 1

2.3 Support Vector Machine

We also used Support Vector Machine with a linear Kernel implemented using sklearn.linear_model module. Initially the SVM didn't had the regularization parameter as a result of which it tended to over fit the training set. Therefore, we introduced a regularization term C via soft-margin classifier. We used both "linear" and "rbf" kernels and optimized the classifier with various combination of C and alpha. We also used a SGDClassifier with alpha = 0.0001 and penalty = 'l2' with "hinge" loss. We then did a grid search over various combinations of parameter grid to select the best one. The result was following values:

- 'alpha': 1e-05,
- 'n_iter': 10,
- 'penalty': 'elasticnet'

We're able to achieve a 96.62% accuracy using these parameters. The full results of the grid search with 3-fold cross validation can be found in the ipython notebook mentioned in the References section. Classification report for SVM is given in Figure 3.

Classification report [SGD classifier]

Class	precision	recall	f1-score	Support
business	0.95	0.92	0.94	102
entertainment	0.99	0.92	0.95	84
politics	0.92	0.99	0.95	78
sport	0.99	0.98	0.99	103
tech	0.93	0.97	0.95	78
avg / total	0.96	0.96	0.96	445

Confusion Matrix

business	[94,	0,	5,	0,	3]
entertainment	[3,	77,	2,	0,	2]
politics	[0,	0,	77,	1,	0]
sport	[1,	0,	0,	101,	1]
tech	[1,	1,	0,	0,	76]

Figure 3: Classification report and Confusion matrix for SVM with SGD training

Why SVM works well with Text classification?

- SVM protects overfitting and hence works well in high dimensional space like sparse matrix used in bag-of-words model.
- One news articles vectors are sparse and hence SVM's works well.
- The problem of news article classification is mostly linearly separable.
- There are few irrelevant features in the bag of words model and every single features contributes to the class.

2.4 K-means clustering (Unsupervised approach)

We also used K-means to cluster the documents into 5 categories and use MDS technique to plot the news articles onto a 2-D space. We used 30 % of news data in this approach as compared to 80-20 in the supervised approach for better visualization of the Dendrogram and MDS plot. Number of iteration was set to default 100 and we observed that the K means converged in 16 iterations. The scores for the various parameters are as follows:

```
Homogeneity: 0.530
Completeness: 0.582
V-measure: 0.555
Adjusted Rand-Index: 0.423
Silhouette Coefficient: 0.011
```

For plotting the MDS we have used cosine similarity as a distance metric to measure the Distance b etween documents. Cosine similarity is measured against tf-idf matrix and can be used to generate measure of similarity between articles. The MDS plot can be found in section 3 Results.

We also did hierarchical clustering on the data, Ward clustering algorithm which uses single link distance between clusters as a distance measure. We have used the cosine distance matrix to calculate distance matrix which is further used to plot the Dendrogram.

2.5 Performance Metric

It is of crucial importance to categorize the news articles into the 5 categories as the data is relative ly subjective. Like a Tech story more focused on Business can fall in Business category. Accuracy of classifying the articles correctly is the primary metric for our classification process.
We'll have evaluated all our algorithms based on accuracy.

Metrics like precision, Recall, F1 macro and F1 micro score can be used where classifying incorrect r esults have huge consequences like spam filtering. Classifying a non-spam email as spam would be lead to missing information for the user.

3 Results

We have executed 10 – fold cross validation on Naïve Bayes and SVM for calculating the average ac curacy of the models. Average accuracy for the models is 80.56 % and 95.64 % respectively. Figure 4 shows a comparison of all the models considered in this task with Accuracy as performance metric.

Sr. No.	**Model**	**Accuracy (%)**	**Parameters**
1	Decision Tree	80.22	Entropy model
2	Logistic Regression	92.80	Default
3	Naïve Bayes	84.26	Alpha = 1
4	Naïve Bayes	87.63	Alpha = 0.5
5	SVM (SGD training)	95.50	loss='hinge', penalty='l2 ',alpha=1e-3, n_iter=5, r andom_state=42
6	SVM (SGD training)	96.62	'alpha': 1e-05, 'n_iter': 10, 'penalty': 'elasticne t'
7	SVM	96.40	C = 10 , Kernel = linear
8	Random Forrest	86.74	Trees = 10
9	Random Forrest	94.38	Trees = 100

10	Adaboost	86.29	N = 500
11	Gradient Boosting	96.85	N = 100

Figure 4: Comparison of various classifier with Accuracy as performance evaluation metric

```
Homogeneity: 0.606
Completeness: 0.654
V-measure: 0.629
Adjusted Rand-Index: 0.493
Silhouette Coefficient: 0.017
```

Figure 5: K means scores with k = 5

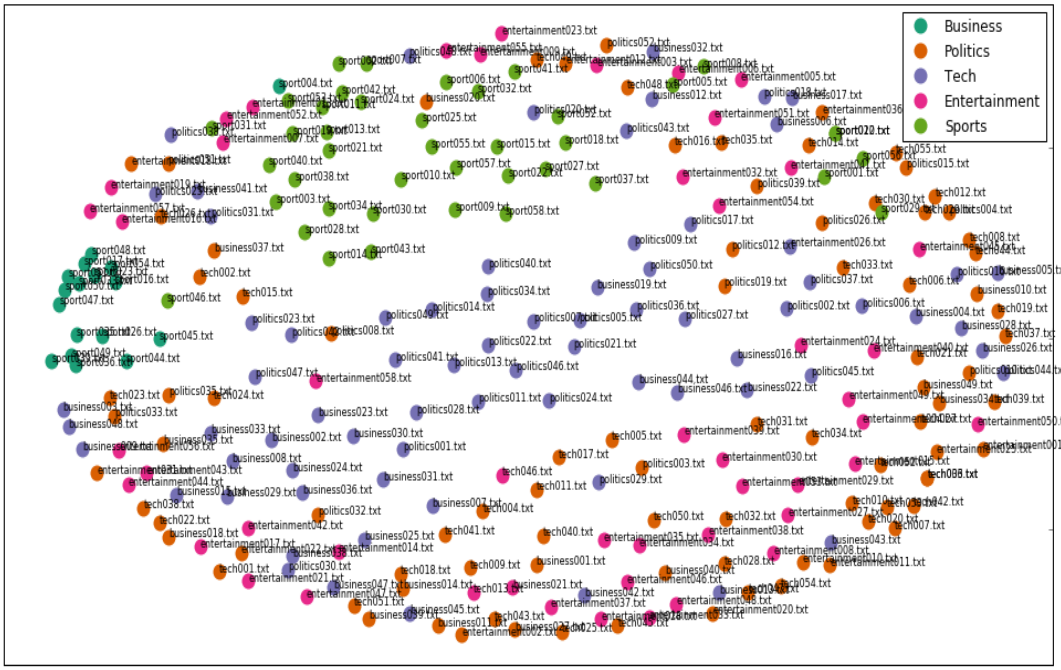

Figure 6: MDS plot for K-means clustering

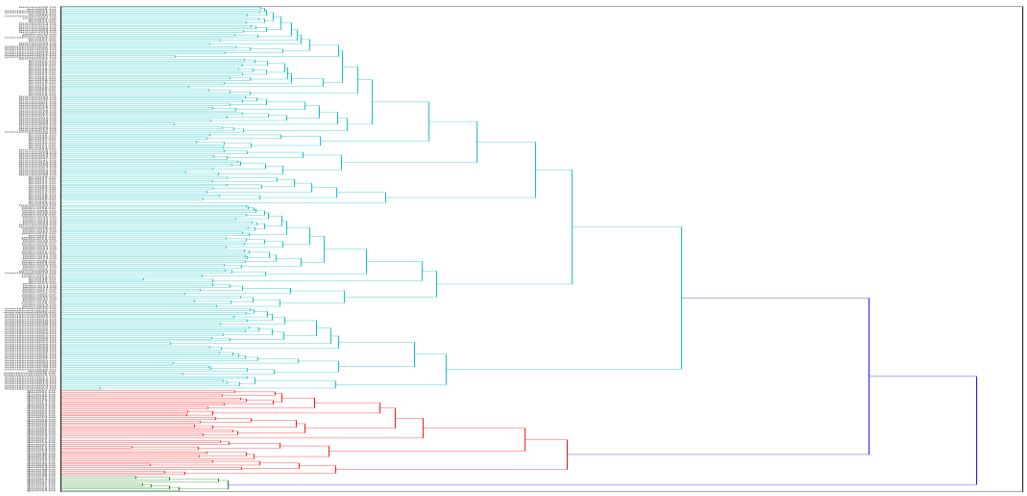

Figure 7: Dendrogram. Ward agglomerative clustering

4 Conclusion and Future work

Future work may involve training the model on BBC datasets and testing it on Reuters dataset or any other similar news channel. The accuracy figures in current analysis are high because the testing and training sets belong to the same domain BBC. Language use, words and other lexical features vary across different news channel and it would be interesting to see the results. The model can also be extended to various other application like Spam filtering, Authorship Attribution, Genre classification etc. that requires bag-of-words model for classification process. We also hope to improve our classifier accuracy by cross domain training (train on BBC, test on Reuters or TEDx) and deploy the model over a web to be used by general users as a classification tool.

References

[1] D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006

[2] Learning to Classify Text, NLTk.org/book

[3] http://mlg.ucd.ie/datasets/bbc.html

[5] Sklearn, http://scikit-learn.org/stable/user_guide.html

[6] http://machinelearningmastery.com/

[7] http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf

[8] Introduction to Information Retrieval, NLP Stanford

[9] https://github.com/kushalkokje/CSCI-B-659-Applied-Machine-Learning/blob/master/News-Articles-Classification/AML-News-classification.ipynb