News Headlines Classification

Rohit Kumar
PID: A53275464
r2kumar@eng.ucsd.edu

Sujith Chappidi
PID: A91021513
jchappid@ucsd.edu

Yuvaraj Kakaraparthi
PID: A53278224
ykakarap@eng.ucsd.edu

ABSTRACT

This paper deals with the problem of classifying different news articles based on text analysis of their headlines. We begin by identifying and exploring the salient features of the data set. Then we use different modeling techniques (Logistic Regression, SVM, Random Forest) and feature construction methods to accurately classify the different news articles. A final analysis of the different methods and our inference for short texts is presented.

1. INTRODUCTION

1.1 Dataset

The dataset used for this assignment is a News Aggregator Dataset from the Machine Learning repository of University of California Irvine.[3] The dataset is a collection of references to news pages collected from a web aggregator in the period from 10-March-2014 to 10-August-2014. The resources are grouped into clusters that represent pages discussing the same story. The dataset contains a total of 422937 news pages. Each data point (news page) in the dataset has the following information:

- Numeric ID
- Title of news article
- URL of the news article
- PUBLISHER of the news
- HOSTNAME: url of the host
- STORY: Alphanumeric ID of the cluster that includes news about the same story.
- TIMESTAMP: Approximate time the news was published, as the number of milliseconds since the epoch 00:00:00 GMT, January 1, 1970
- CATEGORY of the news article
  1. b = Business
  2. t = Science & Technology
  3. e = Entertainment
  4. m = Health

CSE 258, 2018. University of California San Diego
.

1.2 Exploratory analysis

The dataset contains articles from 4 categories namely BUSINESS, ENTERTAINMENT, SCIENCE & TECHNOLOGY and HEALTH. There are 152469 Entertainment articles, 115967 Business articles, 108344 Science & Technology and 45639 Health articles in the dataset.

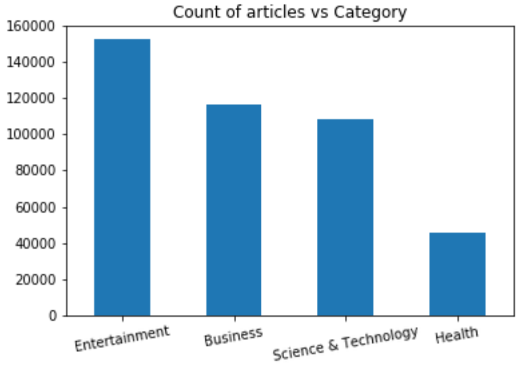

Figure 1: Article Count vs Category

There are a total of 10985 unique publishers in the dataset with 'Reuters' being the top publisher having 3902 publications. The Figure 2 shows the number of articles published by the top 50 publishers.

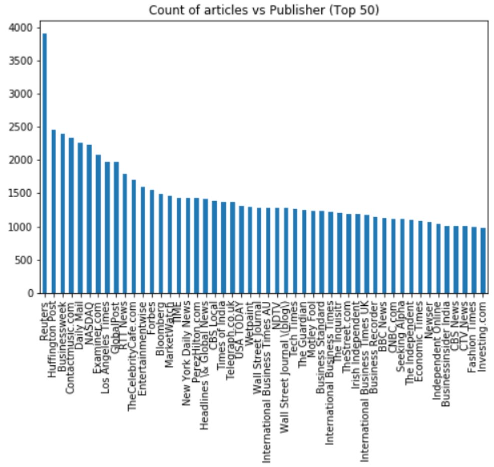

Figure 2: Article Count vs Publisher

Taking a look at the distribution of articles in Figure 3, among the 4 categories for each publisher shows that each publisher in general publishes more articles of one kind over the other. For example, *Reuters* publishes business articles significantly more than other kinds of articles and similarly *Entertaintmentwise* publishes significantly more entertainment articles.
This is expected because in general different publications tend to publish articles of similar kind.

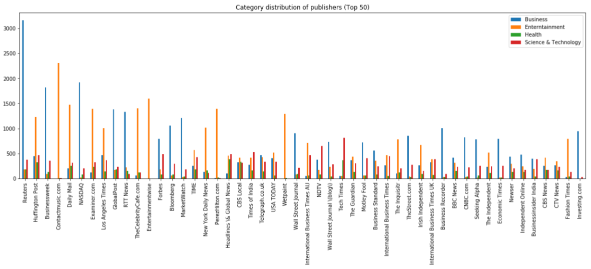

Figure 3: Category Distribution for Publishers

Each data point in the dataset comes with headline of the article. The headline is a short description of the article that captures the context on an article.
Word clouds representing the distribution of words in each of the categories is shown in the Figure 4, Figure 5, Figure 6 and Figure 7. Looking at the top words that occur in each of the 4 categories, it shows that there is very little overlap in the words that are commonly used in the 4 categories.

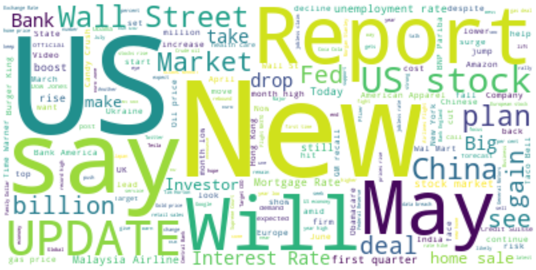

Figure 4: Word Cloud of Business Category

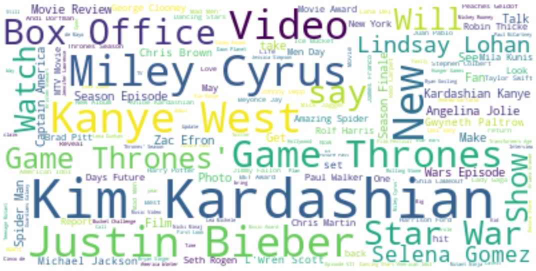

Figure 5: Word Cloud of Entertainment Category

2. PREDICTIVE TASK

The task is going to be to predict the category a news article belongs to given information about the article (like the title of the article). News classification has been the core of classification problems. It has uses cases ranging from user personalization to clustering and categorization.
In this assignment we will be predicting which among the 4 categories of Business, Entertainment, Science & Technology and Health does an article belong to given some information about the article like Title of the article, publisher, etc.

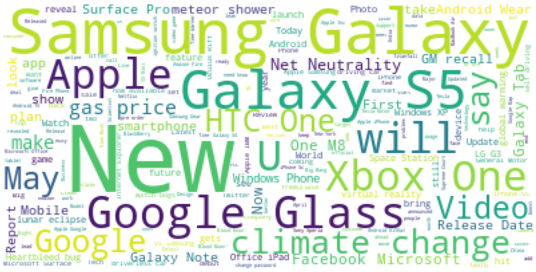

Figure 6: Word Cloud of Science & Technology Category

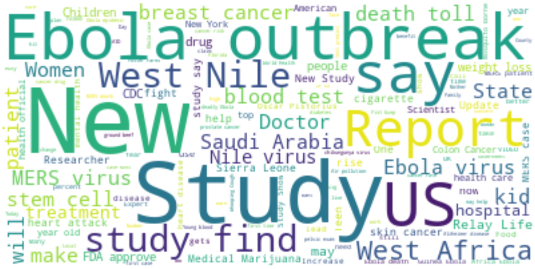

Figure 7: Word Cloud of Health Category

2.1 What is the predictive task?

As we already observed that the 4 categories do not share a lot of words in common (from the word clouds) we will be using the title of the article to make predictions about the category of an article.
Since the title of the article is a sequence of words of variable length we need an approach to convert the title of any length to a vector of fixed length. In this assignment we will be using 2 methods to generate the vector representations for the title of an article.

1. Bag-of-words approach. In Bag-of-words approach a title will be represented by a vector of token counts where each token represents a unique word in the dataset.
2. Averaging the vector representation of individual words. Word embeddings are feature learning techniques in Natural Language Processing (NLP) in which words are mapped to vectors of real numbers.

2.2 Data Pre-processing

Before the title is vectorized every data point (i.e word in the title) is pre-processed sequentially.

1. Converting all characters to lower case.
2. Remove all special characters such as {% ! # $ ˜ : ; ? @}
3. Replace all numbers in the text with the word 'DIG'.

Stemming is not performed for the following reason:
There can be words like 'ratings' and 'rate' which could possible map to Entertainment and Business respectively but

the stemmer results of both 'ratings' and 'rate' will be 'rate'. This could lead to possible loss of predictive words.

2.3 Feature Selection

In Bag-of-words approach each title is represented by a vector of token counts of the unique words in the dataset.This approach is also referred to as CountVectorizer approach in this report.
For the word embedding approach we will be using Google's pre-trained word2Vec embeddings[1]. Google's Word2Vec embedding[1] gives a 300-dimensional vector representation for every word. The title, which is a collection of words, is then converted to a 300-dimensional vector by averaging over the 300-dimensional vectors of each of its words.

3. MODEL

We have explored the following methods with different feature vector representations.

1. LDA (Latent Dirichlet Allocation)
2. Multinomial Logistic Regression
3. Support Vector Machines(SVM)
4. Naive Bayes (NB)
5. Random Forest

We have selected the popular classification algorithms like Multinomial Logistic Regression, Support Vector Machines and Naive Bayes and have compared their performances with respect to different feature vector representations i.e Word2Vec and CountVectorizer features.

3.1 Feature Vector

For training the model for all the above methods, we have labeled the 4 categories that we have, in the following way:

1. 'b' - 0
2. 'e' - 1
3. 'm' - 2
4. 't' - 3

Word2Vec is a pre-trained model that includes word vectors for a vocabulary of 3 Million words and phrases that has been trained on roughly 100 billion words from a Google News dataset[1]. The vector length is 300 features. This feature vector was chosen as it was believed to give a pretty accurate representation of a particular word. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Word2Vec doesn't include some stop words like 'a', 'and', 'of' are excluded, but others like 'the', 'also', 'should' are included. Word2Vec also takes care of mispelled words. Ex - it includes both 'mispelled' and 'misspelled' - the latter is the correct one.

Example of Word2Vec Vectors.

$Word2Vec_{King}$ + $Word2Vec_{Woman}$ - $Word2Vec_{Man}$ is most similiar to the $Word2Vec_{Queen}$

The vectors for a particular word can be obtained from Word2Vec. We, however need the vector for a particular document or a headline in our particular case. To obtain the vector representation of a particular document or a headline, we have taken the average of the vectors of all the words/tokens (without punctuations) in a particular headline.

CountVectorizer has been imported from sklearn library in Python. CountVectorizer gives us a vector of a document in terms of all the unique words that occur in the entire corpus. The vector of a particular document has the respective counts for all the corresponding word while forming the Vector for a particular document.

In the particular case of classification that we have, we have seen that the overlap of words between any two categories is very low, which means that the classification is made simple by using the most simplistic methods rather than trying to understand features using complex algorithms.

The tokens for the LDA Model have been formed on removing punctuations and stopwords. Also, stemming was implemented on these tokens to form the bag-of-words corpus.

3.2 Train,Validation,Test Set

The full dataset, after shuffling, has been divided in the following way. Train set consists of the initial **300,000** entries of the full dataset. The validation set was selected as the entries from **300,000 to 350,000th** entries. The remaining entries were chosen as the test set.

3.3 Baseline Model

Baseline Model that we chose is as follows. From the initial data analysis that we did earlier, we could see that particular publishers usually published a lot of headlines of a few particular categories. We prepared a dictionary of publishers mapped to the category that the publisher published the most.

The category that a headline was categorized to was decided by the publisher only. The most published category by a publisher was chosen as the category for that particular headline. The accuracy obtained for the baseline model on the test set is **63.7807%**

3.4 Performance of the Models

From the accuracies that have been reported in the Table 1, we can see that the accuracies are significantly better for the CountVectorizer feature vector than the Word2Vec feature vector.

Another observation, as evident from Table 1 is that the accuracy for the LDA model is extremely low. Probabilistic models such as LDA exploit statistical inference to discover patterns of data. The model infers parameters from observations. The accuracy of statistical inference depends on the number of your observations. LDA models a document as a mixture of topics and the words are drawn from each topic later. Since the documents that we are using are just headlines, the LDA model doesn't have enough observations to generate a good enough model.

Example of the topic that we got from the LDA topics- Topic: 0.020*"google" + 0.017*"apple" + 0.013*"samsung" + 0.011*"microsoft" + 0.011*"galaxy" This topic looks predominantly like Science and Technology. Similarly, we got words for other topics as well. However, as we look beyond just the top 5 words we find words like "thrones" for this topic, which has almost nothing to do with Science & Technology. It is these kind of words that can be the reason why LDA gives us such low accuracy.

From the LDA model that has been generated, the 4 topics have been formed but the accuracy is very bad because of the problem of short descriptions for the categories.

Table 1: Accuracies

Feature Vector	Model	Test Set Accuracy
CountVectorizer	Naive Bayes	92.5254%
Word2Vec	Naive Bayes	76.6801%
CountVectorizer	Multi. Log. Regression	94.7265%
Word2Vec	Multi. Log. Regression	85.2138%
CountVectorizer	SVM	94.569%
Word2Vec	SVM	85.9330%
CountVectorizer	Random Forest	86.8225%
Word2Vec	Random Forest	86.7589%
-	LDA	55.438%
-	Baseline	63.7807%

From the initial data analysis that we did, we could see that the overlap of words between any two categories is very less. Since this would mean that the datapoints are largely separated we can see that the simpler methods like, Logistic regression and SVM perform very well (as shown in the table). Logistic Regression minimizes the logistic cost function while SVM minimizes the number of misclassifications. Both these methods perform significantly better because of the nature of the dataset we have.

4. LITERATURE

4.1 Dataset

The dataset used for this assignment is a News Aggregator Dataset from the Machine Learning repository of University of California Irvine.[3]. This news category dataset has been curated by the Artificial Intelligence Lab @ Faculty of Engineering, Roma Tre University - Italy.
Kaggle also provides this dataset on the platform and study of the various approaches revealed that this dataset has been used for primarily 2 tasks: classification problems and clustering problems.
There are many similar datasets pertaining to news articles and headlines. Some of the common ones are listed below.

1. Reuters Newswire Topic Classification (Reuters-21578). A collection of news documents that appeared on Reuters in 1987 indexed by categories.
2. ABC news headlines. News headlines published over a period of 15 years by ABC.
3. 1000 Usenet articles were taken from 20 different newsgroups.

Most of the work on news datasets relates to primary three categories: Classification, Sentiment Analysis, Information Retrieval and Summarization. Recent work has also been focused on new problems such as identifying 'fake news'.

4.2 State-of-art methods

Text classification as a problem has seen implementation of various new breakthroughs. Latent Dirichlet Allocation (LDA)[2] is one of the more famous unsupervised learning approach to understanding the structure and hidden topics behind a text corpus. Recent advances in deep learning techniques have been particularly helpful in solving some of the text analysis problems.

The word2vec method for generating embeddings [4] is a Recurrent Neural Network techniques and more specifically LSTM is one of the more modern techniques being used to find even more patterns within text.

4.3 Comparisons

The models proposed for our classification task use a combination of the state-o-art techniques along with the older methods. We are obtaining the word embeddings through the deep learning technique of word2Vec. This is in turn used as a feature for simpler classification models. We are not using any neural network based approach for the training part primarily based on our initial analysis that the dataset is highly clustered with very small of words occuring across headlines of different categories. This assumption would not hold true if entire body of articles are used for classification and neural networks based approach are bound to be more suitable there.

5. RESULTS

The accuracy for the classification task for the various models and feature representations in Table 1. From the values observed, we see that there are 3 distinct clusters of model accuracy.

1. Models with features constructed using Count Vectorizer. The Naive Bayes, Mutinomial Logistic Regression, SVM and Random Forest models with this feature vector construction had accuracies of 92.52%, 94.72%, 94.57% and 86.82% respectively.
2. Models with features constructed using word2vec. The Naive Bayes, Mutinomial Logistic Regression, SVM and Random Forest models with this feature vector construction form the next cluster with accuracies of 76.68%, 85.21% , 85.93% and 86.76% respectively.
3. Model with classification done through topics suggested by LDA. This model performed the worst with an accuracy of 55.44%.

5.1 Comparison of Model Performance

5.1.1 On basis of the feature vectors

As we had stated earlier that given a phrase in a headline, if that is a common phrase in more than one of the four categories of headlines, Word2Vec will construct the feature vectors such that the word is equally probable to belong to either of those categories. Hence, this can sometimes predict the wrong category.
E.g.: A phrase such as '1 billion' or 'billions' could well belong to business or entertainment.

Count Vectorizer on the other hand simply counts the number of occurrences of a particular word in the given news headline. Hence, it does not suffer from the above misclassification.
The analysis stated here is valid for the case where we have short headlines containing few words.

5.1.2 On basis of Models Used

From the word clouds generated for the four categories of news headlines, we made the observation that there is a very small overlap of words between any two categories. Hence the news headlines are well differentiated in terms of their word usage with very few difficult cases. Since Logistic Regression tries to maximize the score for the most confident classifications while SVM focuses on the data points at the classification boundary, Logistic regression provides us a better accuracy.

5.2 Parameters

The parameters for the various models which we used for comparison can be summarized as follows.

1. Multinomial Logistic Regression. solver = 'lbfgs'. multi_class='multinomial'
2. SVM. The kernel use was 'linear' and the regularization parameter 'C' was set to 1. These parameter values worked well because the news headlines dataset was already highly clustered with few overlaps.
3. Random Forest. For CountVectorizer features 'n_estimators' - number of trees in the forest - is set to 1000 and 'max_depth' - maximum depth of a tree - is set to 200. For Word2Vec features 'n_estimators' is set to 150 and 'max_depth' is set to 32.

In conclusion, we were able to establish that the classification problem for short text sentences was best dealt with the simple models such as Logistic Regressions and SVM. The feature construction for the sentences was also studied and we can conclude that for a curated dataset with clear differentiation between clusters, the count Vectorizer method for constructing word embeddings gives better results than the embeddings generated through word2Vec.

6. REFERENCES

[1] Google's pre-trained word2vec model.

[2] M. J. David Blei, Andrew Ng. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 2003.

[3] D. Dheeru and E. Karra Taniskidou. UCI machine learning repository, 2017.

[4] G. C. J. D. Tomas Mikolov, Kai Chen. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013.