Comparative Analysis on Topic Modeling and Classification on Short Text Dataset

Fatema-E- Jannat
Dept. of Electrical Engineering
University of South Florida
Tampa, Florida, USA
fjannat1@mail.usf.edu

***abstract*—**The traditional methods of topic modeling such as pLSA, LDA has shown a remarkable performance already.. But these methods are implemented on relatively long text based dataset. So my plan is to do a comparative analysis on a text dataset which contains short text and by comparing different methods to find the best model. For the text classification problem implementing Naive Bayes, Regression Analysis, CNN in supervised method, I found that CNN showed the best accuracy among those which is around 65%. For the topic modeling problem, I implemented LDA and NMF method to observe the performances and measured the performance by comparing the coherence value.

Keywords—classification, Naive-bayes, Random-forest, CNN, LDA, NMF, coherence

I. Introduction

The traditional method of text modeling such as pLSA, LDA have achieved remarkable outcome implementing on long text based dataset. In recent days, from social network posts, tweets, messages short texts are being produced which are typically contain less than 15 words and this makes the short text analysis difficult compared to relatively large texts. That's why I focus on the analysis of short text based dataset.

I will implement different classification methods to classify a news headlines dataset to predict the category of the news. Then topic modeling methods will be implemented to categorize the headlines based on their genre.

In this project I covered both the text classification and topic modeling. In section II, the development plan will be presented. Section III will give an overview of the data acquisition and data information. In Section IV a brief description will be proved in text preprocessing. In section V, the model implementation will be described. Finally in section VI, by comparing different methods the overall performance will be discussed.

II. Development Plan

Step 1: In the initial step of this project is to acquire dataset, load and read.

Step 2: After data acquisition, the next step is to preprocess the data. Since the acquired dataset is in raw text format which is not readable by the machine so it needs to be preprocessed to convert the dataset into an understandable format for the machine.

Step 3: After text preprocessing different classification and topic modeling methods will be implemented and performance will be measured for each of those methods.

Step 4: Then a comparison will be made among those methods and observing the comparison the best model will be found for the classification or topic modeling for the acquired dataset.

Fig. 1. Flow diagram describing the overall development plan

III. Data Acquisition

For this project I have used two datasets. Both of them are collected from Kaggle.

The dataset-1 consist of over 200k news headlines and 6 attributes and it is in json format. Among those 6 attributes only "category" and "headline" is considered for the classification task.

The dataset can be found here, https://www.kaggle.com/rmisra/news-category-dataset.

The dataset-2 consists of 1,103,663 instances with publish_date and headline_text attributes. This dataset is used for topic modeling task. The news headlines are taken from reputable Australian news source ABC (Australian Broadcasting Corp.) (http://www.abc.net.au/).

The dataset can be found here, https://www.kaggle.com/therohk/million-headlines

IV. Text Preprocessing

Text Preprocessing is the method of converting the text data into a machine readable format. So the human languages are converted into a machine analyable form after text preprocessing.

There are several library exists for text preprocessing nowadays. Among those, NLTK (Natural Language Toolkit) is one of the most popular NLP library for text data preprocessing in Python language.

There are some basic text pre-processing methods which includes lower-casing, removing punctuations, removing stop words, tokenization, stemming, lemmatization.

The advanced level of text pre-processing includes Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Words, Word Embedding and so on.

A) Basic Text Preprocessing Steps:

Lower Casing: Lowercasing the text reduces the size of the vocabulary. It helps to ignore multiple copies of the same word. "Word", "WORD", "word" will be taken as a single word after implementing the lower casing method.

Removing Punctuations: Since punctuations do not add any information to the text, so removing punctuations helps to reduce the data size.

Removing Stop Words: Stop words are commonly used words such as "is", "was", "the", "on" which contain low information, so removing these stop words will significantly reduce the data size without putting any negative impact on the performance.

Tokenization: Tokenization means to split the sentence into a sequence of words.

Stemming: Stemming is the method of removing suffices from the words. For example, "played", "plays" will be transformed to "play".

Lemmatization: Lemmatization transforms a word to its root word which is more effective than stemming method.

B) Advanced Text Preprocessing Steps:

Bag of Words: This is the method which represents the text that explains the presence of words within the document. For this, a vocabulary list is created with known words and then make a measurement for the

presence of known words for any document.

Word Embedding: Word embedding is the process of representing any text to vector form. In this way the same type words will have minimum distance for their vectors.

TF-IDF: This is a method which identifies the importance of any word to a document in a text dataset and give weightage based on that. For instance, it gives lower weight on “do”, “use” and gives higher weight on “thanks”.

C) Data visualization before and after text preprocessing:

	category	headline
0	CRIME	There Were 2 Mass Shootings In Texas Last Week...
1	ENTERTAINMENT	Will Smith Joins Diplo And Nicky Jam For The 2...
2	ENTERTAINMENT	Hugh Grant Marries For The First Time At Age 57
3	ENTERTAINMENT	Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4	ENTERTAINMENT	Julianna Margulies Uses Donald Trump Poop Bags...

Fig.2: The dataset-1 is shown here which is before preprocessing but removing other attributes which will not be considered in this analysis

	headline_text
0	aba decides against community broadcasting lic...
1	act fire witnesses must be aware of defamation
2	a g calls for infrastructure protection summit
3	air nz staff in aust strike for pay rise
4	air nz strike to affect australian travellers

Fig.3: The dataset-2 is shown here which is before preprocessing but removing other attributes which will not be considered in this analysis

	category	headline
0	CRIME	mass, shooting, texas, last, week, ,, tv
1	ENTERTAINMENT	smith, join, diplo, nicky, jam, world, cup, 's...
2	ENTERTAINMENT	hugh, grant, marries, first, time, age
3	ENTERTAINMENT	jim, carrey, blast, 'castrato, ', adam, schiff...
4	ENTERTAINMENT	julianna, margulies, us, donald, trump, poop, ...

Fig.4: The dataset-1 is shown here which is after applying preprocessing methods.

	headline_text
0	[aba, decid, against, commun, broadcast, licenc]
1	[act, fire, wit, must, be, awar, of, defam]
2	[a, g, call, for, infrastructur, protect, summit]
3	[air, nz, staff, in, aust, strike, for, pay, r...
4	[air, nz, strike, to, affect, australian, travel]

Fig.5: The dataset-2 is shown here which is after applying preprocessing methods.

V. Model Implementation

Here I have implemented text classification and topic modeling on two dataset.

I have split the whole dataset into two separate parts, training and testing. For training I have used 60% of the dataset and for testing I have used 40% of it. For the sake of this analysis and for the time and memory constraints I had to reduce the whole dataset-1 to 10,000 instances. For these 10,000 instances the dataset-1 has 26 unique categories. I have used Python language and their library such as Keras, Scikit-learn to implement different classification and topic modeling methods

For the classification analysis I have implemented three models, named Naive Bayes, Random Forest, MLP, CNN.

For the topic modeling analysis I have implemented LDA and NMF..

VI. Experiments and Results

A) Text Classification Analysis:

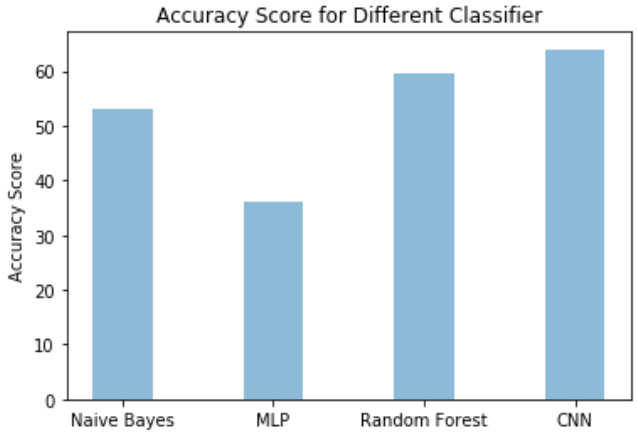

Fig.6: Accuracy score for different classifier for Dataset-1

B) Topic Modeling:

i)LDA (With bag of words):

Topic: 0 Words: 0.017*"market" + 0.013*"tasmania" + 0.012*"price" + 0.012*"open" + 0.011*"share" + 0.010*"victoria" + 0.009*"island" + 0.009*"christma" + 0.008*"storm" + 0.008*"campaign"
Topic: 1 Words: 0.045*"new" + 0.013*"council" + 0.013*"chang" + 0.012*"health" + 0.011*"say" + 0.009*"school" + 0.009*"indigen" + 0.008*"servic" + 0.008*"meet" + 0.008*"worker"
Topic: 2 Words: 0.036*"polic" + 0.022*"man" + 0.019*"die" + 0.019*"crash" + 0.018*"car" + 0.015*"death" + 0.013*"investig" + 0.012*"woman" + 0.011*"driver" + 0.011*"attack"
Topic: 3 Words: 0.030*"man" + 0.026*"court" + 0.024*"charg" + 0.020*"year" + 0.020*"murder" + 0.019*"interview" + 0.018*"face" + 0.015*"found" + 0.013*"accus" + 0.012*"sex"
Topic: 4 Words: 0.026*"govern" + 0.019*"nsw" + 0.017*"rural" + 0.014*"qld" + 0.014*"say" + 0.014*"state" + 0.013*"nation" + 0.011*"labor" + 0.010*"protest" + 0.010*"support"

Table.1: Resultant Keywords for each topic and the weightage(importance) of each keyword while performing LDA using bag of words on Dataset-2

ii)LDA (With TF-IDF):

Topic: 0 Word: 0.021*"trump" + 0.019*":" + 0.010*"australia" + 0.009*"day" + 0.007*"live" + 0.007*"world" + 0.006*"test" + 0.006*"energi" + 0.006*"win" + 0.006*"juli"
Topic: 1 Word: 0.022*"man" + 0.015*"polic" + 0.014*"charg" + 0.012*"murder" + 0.011*"woman" + 0.011*"crash" + 0.009*"court" + 0.009*"car" + 0.008*"found" + 0.008*"drum"
Topic: 2 Word: 0.010*"turnbul" + 0.009*"elect" + 0.007*"labor" + 0.006*"marriag" + 0.006*"abus" + 0.006*"$" + 0.006*"liber" + 0.006*"abbott" + 0.006*"malcolm" + 0.006*"royal"
Topic: 3 Word: 0.015*"news" + 0.011*"abc" + 0.011*"rural" + 0.009*"nrl" + 0.008*"christma" + 0.008*"sport" + 0.008*"nation" + 0.007*"friday" + 0.007*"septemb" + 0.007*"peter"
Topic: 4 Word: 0.008*"grandstand" + 0.007*"us" + 0.006*"islam" + 0.006*"kill" + 0.006*"terror" + 0.005*"refuge" + 0.005*"syria" + 0.005*"australian" + 0.005*"australia" + 0.005*"attack"

Table.2: Resultant Keywords for each topic and the weightage(importance) of each keyword while performing LDA using TF-IDF on Dataset-2

Method	Coherence Score
LDA(Bag of words, 10 topic)	0.21
LDA(Bag of words, 35 topic)	0.27
LDA(Bag of words, 45 topic)	0.28
LDA(TF-IDF, 10 topic)	0.33
LDA(TF-IDF, 35 topic)	0.47
LDA(TF-IDF, 45 topic)	0.84

Table.3: Respective coherence score for different approaches

The experimental setup was divided into two parts, one is classification and the other one is topic modeling. In the classification task, after applying several traditional methods the CNN classifier was able to achieve the

highest accuracy score after 10 epos which is around 64%, the accuracy score comparison is shown in fig.6. For the CNN classifier “relu” activation function was used, for the optimizer “adam” was used.

For the topic modeling LDA is implemented using Bag of words and using TF-IDF. For each of those cases the resultant keywords for each topic with their corresponding weightage is shown in table.1 and table.2 .

For evaluating LDA coherence is measured. Coherence score defines the relative distance between words for topics. For the case of LDA-1 (using bag of words) I got 0.28 coherence score and For the case of LDA-2 (using TF-IDF) I got around 0.84 coherence score after setting the topic number as 45. I set this topic number after some trial and error with the numbers. Table.3 shows the comparative analysis on different topic modeling approaches.

VII. Conclusion

Using datasets from kaggle which consists of short texts I executed a comparative analysis using existing text classification and topic modeling approaches to observe how they perform on short texts. My finding from the comparisons is that for the classification the CNN performs better than any other classifiers and was able to achieve 64% accuracy and for the topic modeling using TF-IDF it got 0.84 coherence score whereas for bag of words it got only 0.28 coherence score.