A Study on Natural Language Processing approaches for Text2Image using Machine Learning Algorithms

Syed Muzamil Basha, Priyadharsheni JM, Sajeev Ram & N.Ch.S.N.Iyengar*
Department of Information Technology
Sri Krishna college of Engineering and Technology, Coimbatore-641008.
*Sreenidhi Institute of Science and Technology, Yamnampet,Ghatkesar, Hyderabad (T.G),

Abstract

Now a days, Natural Language Processing (NLP) tasks are everywhere around us like, suggest in search, automatic Gmail replies, machine translation. The challenges in the field of NLP, is having lots of text data in the form of reviews available in the online platforms like Twitter and Facebook. It requires huge man power in making use of this data in deriving useful insights from the text data. The better way to reduce the human effort and reduce the time taken in making the text data ready for decision making in Text2Img representation. The approach followed in this paper is to convert text in to image. Later, apply Machine Learning Algorithms (MLA) to perform prediction based on user choice. In the experiment conducted, we constructed a word cloud to understand the theme (Love and Time) of the artist. In addition, to that applied MLA - Random Forest (RF) and achieved 78% classification accuracy on the dataset considered

Keywords:- Natural Language Processing, Machine Learning Algorithms, Word Cloud, Random Forest, Classification Accuracy.

I.INTRODUCTION

In this paper, our discussion is mainly on how to represent pieces of text with vectors. So, that TF_IDF is computed between vectors. Which is used in the popular NLP applications like: text classification [1] or duplicate detection, named entity recognition. There are three main approaches in NLP: one, Rule-Based approaches. In which, regular expressions would be used. Second, making use of Traditional machine learning algorithms. And the last one would be deep learning that has recently gained lots of popularity in NLP. In Rule-Based approach, the context-free grammars shows what would be the rules to produce some words. Now, this context-free grammar is made in use to construct parse of data. Which helps is determining list of non-terminal words. The disadvantage of this approach is to frame the rules manually, this is very time consuming. Usually, rule-based approaches [2] have high precision but low recall. Now, another approach would be to build some machine learning system. Which requires training data (or) corpus with some Label (Supervised Corpus) [3]. On top of which, the probabilistic model are used in deriving the probability of each word in given text input. But generally, these models should be trained with selected parameters and fit the model to data. The similar steps are followed in deep learning approach in training the Convolutional Neural Network (CNN). Deep learning methods perform a vital role in many tasks in NLP. Like, word2vec method which is actually not even deep learning but it is inspired by CNN.

The paper is organized as follows: In Introduction Section, Background knowledge, applications and approaches of NLP is described. In Methodology Section, the steps followed in Text mining and analyzing the insights from Text is presented with all the technical details. In Result and Discussion section, the details of experiments and the outcome is interpreted with the help of equations. In Conclusion and Future work section, the limitation and advantages of the research work with the future direction to address the limitation in the present work.

II.Proposed Algorithm

There are different stages of analysis for that sentence. The first stage, is morphological stage, would be about different forms of words (Part Of Speech (POS)). Then the next stage, syntactical analysis, will be about different relations between words in the sentence. The next stage, once we know some synthetic structures, would be about semantics. So, semantics is about deriving the meaning. The last stage is pragmatics would be of highest level abstraction. Stanford parser is for synthetic analysis that provides different options and has really lots of different models built in. Whereas Gensim and MALLET would be about more high level abstractions. Let us take sentiment analysis as an application area, where one can have a text of review as an input, and produce the class of sentiment as output (Ex: It could be two classes like positive and negative). One can think of text as a sequence of tokens. The process of extracting those tokens, is called Tokenization, and token is like a meaningful chunk of our text. It could be a word (or) sentence. Next is to normalize those tokens using either stemming or lemmatization.

Next is to transform extracted tokens into features for our model. A simple counter features in bag of words manner. In which, each text is replaced by a huge vector of counters. Similarly to preserve local ordering add n-grams. It actually improves the quality of text classification. Replace the counters with Term Frequency – Inverse Document (TF-IDF) [4] values and that usually gives a performance boost.
The steps followed in applying Machine Learning algorithm (Random forest) on the text data are:
1. Created the corpus
2.Clean the Text data in corpus using tm_map()
3. Removing all the English stop words
4. Perform stemming on the corpus
5. Creation of Document Term Matrix (DTM) [5],
6. Estimated TF-IDF values from the corpus.
7. Eliminated the Sparse Terms based on frequencies having confidence level 0.995.
8. Partitioning the data as training (75%) and testing (25%) datasets.
9. Apply Random forest algorithm to obtain confusion matrix [6].
10. Obtained the Accuracy of the model from confusion matrix.

The summary on the dataset considers in the experiments have 824 observations and 20 instances. In which, Text is the name of the instances having the review of the user. The same is presented in Fig. 1

```
> str(prince_orig)
'data.frame':   824 obs. of  20 variables:
 $ X           : int  49 669 78 475 811 478 208 397 714 898 ...
 $ text        : chr  "All 7 and we'll watch them fall\nThey stand in the way of love a
nd we will smoke them all\nWith an intellect an"| __truncated__ "319, 'bout time, come
 in, ow, 319\nTake off your clothes, 319 bet you got a body\nBy God, come on, let me se
e, "| __truncated__ "Don't worry, I won't hurt you\nI only want you to have some fun\nI
 was dreamin' when I wrote this\nForgive me i"| __truncated__ "Prince\nMiscellaneous\n2
020\nThe year is 2020 \nAnd in the club - Love4OneAnother \nStudents dance 2 the heart
b"| __truncated__ ...
 $ artist      : chr  "prince" "prince" "prince" "prince" ...
 $ song        : chr  "7" "319" "1999" "2020" ...
 $ year        : int  1992 NA 1982 NA 2006 NA NA NA NA NA ...
```

```
> glimpse(prince[149,])
Observations: 1
Variables: 7
$ lyrics <chr> "In a room full of harlots and fantasy\nDestiny beckoned us ther...
$ song   <chr> "curious child"
$ year   <int> 1996
$ album  <chr> "Emancipation"
$ peak   <int> NA
$ us_pop <chr> NA
$ us_rnb <chr> NA
```

Fig. 1. Description on Dataset used in the experiment

After all the preprocessing steps applied to the Text review in the dataset, the resulted Text Review is presented in Fig. 2.

```
> str(prince[149, ]$lyrics, nchar.max = 300)
 chr "in a room full of harlots and fantasy destiny beckoned us there curious child on
 the balcony we took the dare careless i was to caress thee yet never regretting the ti
me the joy that we shared  it was meant to be and not a crime  no  it not a crime if me
mory serves us  we will align"| __truncated__
```

Fig. 2. Structure of Text Review after preprocessing

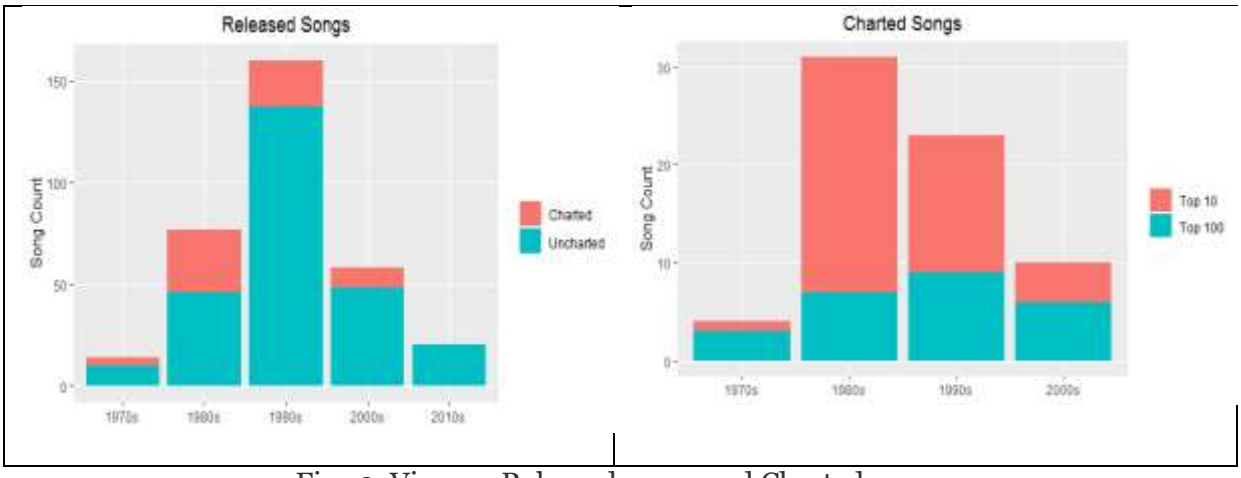

Fig. 3. View on Released songs and Charted songs

In Fig. 3, the categories of charted and uncharted songs with count represent as y-axis is presented. Similarly, on right hand side the top 10 songs for each year is plotted (x-axis).

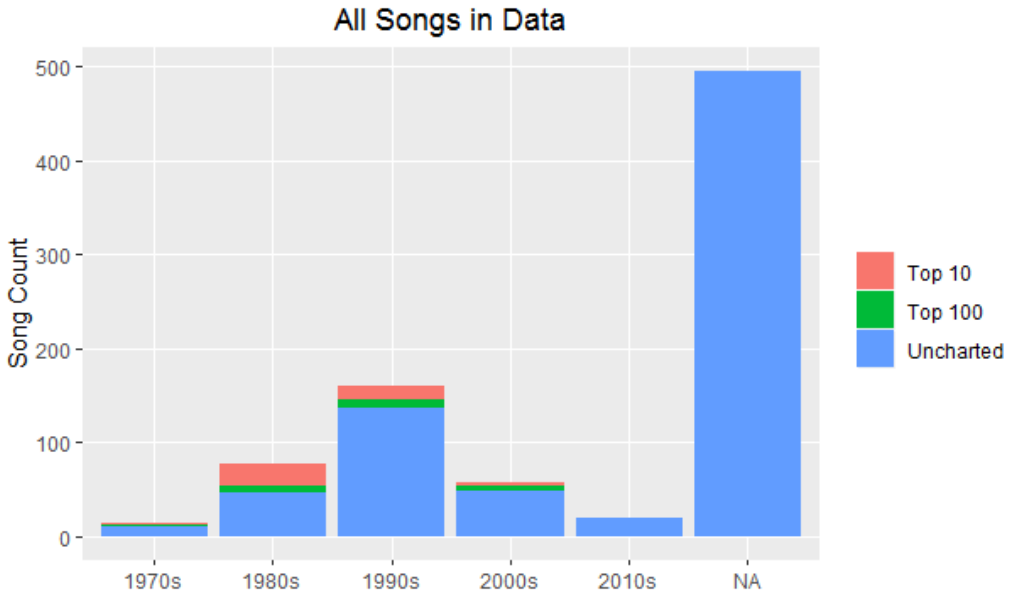

Fig. 4. Different songs in Data

All the categories of classes (Top 10, Top 100 and uncharted) is plotted in Fig. 4 towards findind the popu;arity of the artist from years (1970 to 2010). The details of other attrubutes of the dataset is plotted in Fig. 5.

Tokenized Format Example

word	song	year	peak	decade	chart_level	charted
race	lovesexy	1988	1	1980s	Top 10	Charted
race	my tree	NA	NA	NA	Uncharted	Uncharted
race	positivity	1988	NA	1980s	Uncharted	Uncharted
race	race	1994	NA	1990s	Uncharted	Uncharted
race	sexuality	1981	88	1980s	Top 100	Charted
race	slow love	1987	NA	1980s	Uncharted	Uncharted
race	the rest of my life	1999	NA	1990s	Uncharted	Uncharted
race	the undertaker	NA	NA	NA	Uncharted	Uncharted
race	u make my sun shine	NA	NA	NA	Uncharted	Uncharted
race	welcome 2 the rat race	NA	NA	NA	Uncharted	Uncharted

Fig. 5. Tokenized representation of dataset

After constructing the TDM, each song can be classified based on word count as shown in the Fig. 6

Songs With Highest Word Count

song	chart_level	num_words
johnny	Uncharted	1349
cloreen bacon skin	Uncharted	1263
push it up	Uncharted	1240
the exodus has begun	Uncharted	1072
wild and loose	Uncharted	1031
jughead	Uncharted	940
my name is prince	Top 10	916
acknowledge me	Uncharted	913
the walk	Uncharted	883
the purple medley	Uncharted	874

Fig. 6. Songs with Highest word count

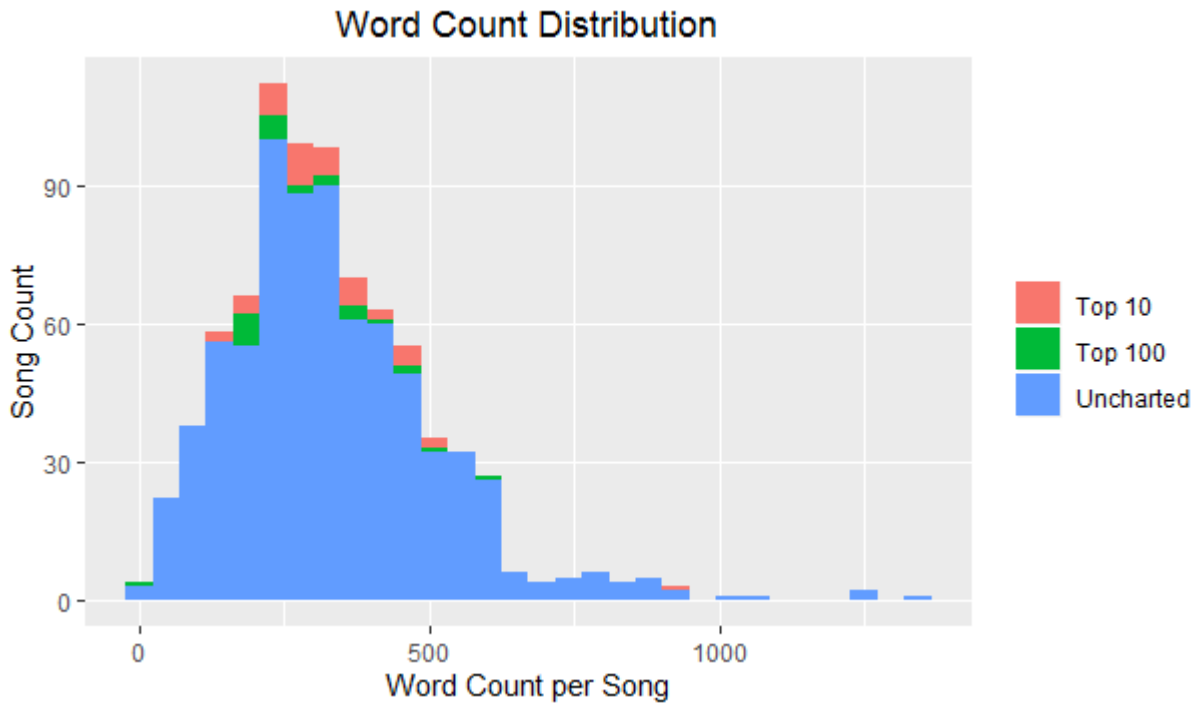

Fig. 7. Word Count Distribution

Later, the distribution of word count [7] on the dataset considered is plotted in Fig. 7. Where all the categories of target variable is considered. This data helps in constructing the wordcloud [8] as shown in Fig. 8.

Fig. 8. Word Cloud of Dataset

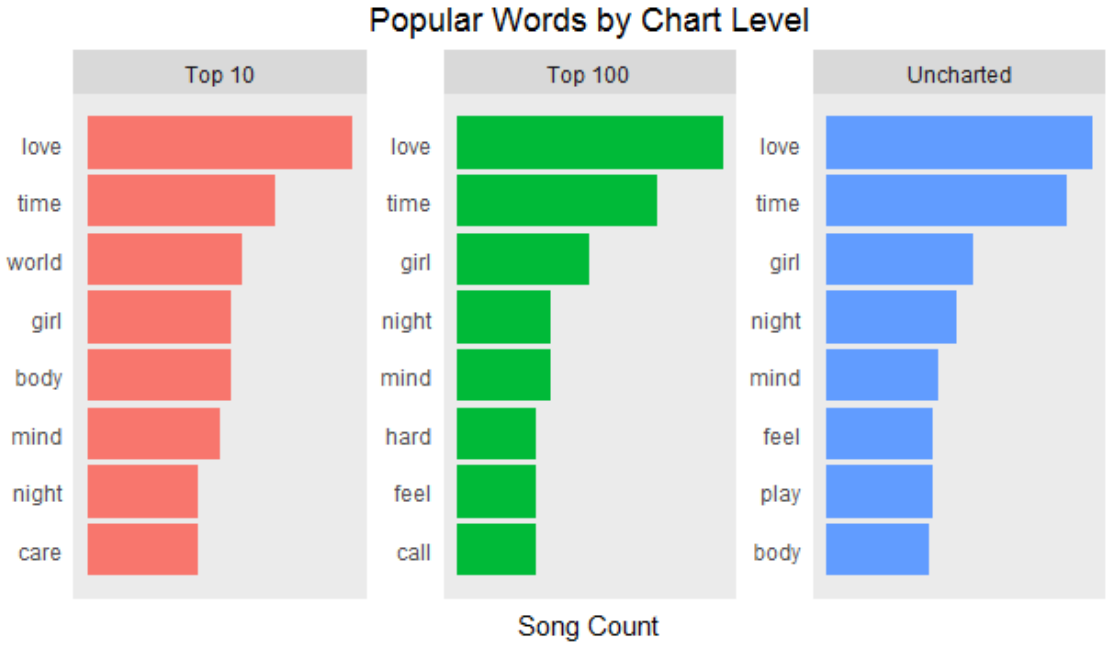

Fig. 9. Popular words by chart Level

The new insight, from the data can be popular words used by chart level. that helps in understanding the theme of the artist (Love and Time) as shown in the Fig. 9. and year-wise in the Fig. 10.

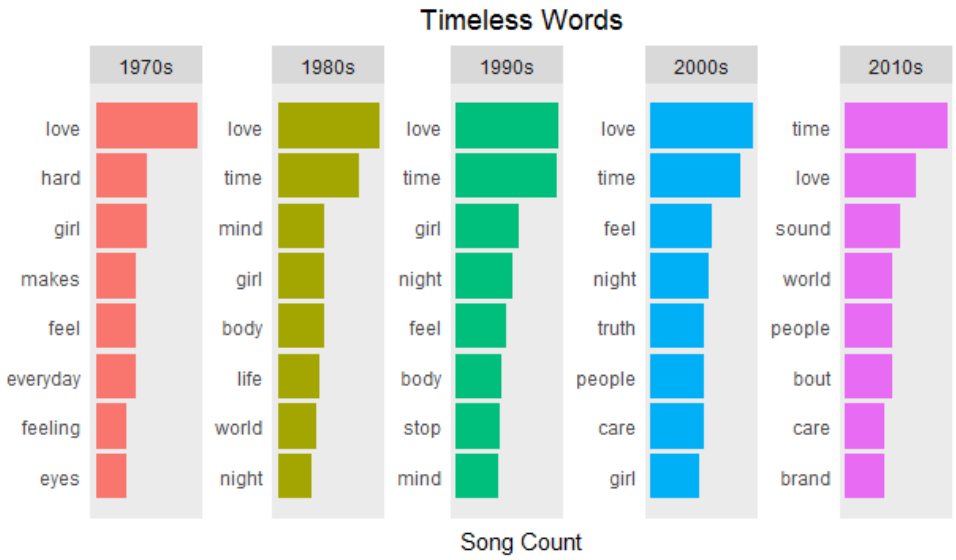

Fig. 10. Timeless words

III. Experiment and Result

To apply Machine Learning algorithms (MLA) for Natural Language processing, the datasets from sklearn package is considered. From which the text data is extracted and TF-IDF is estimated for all the documents in the dataset. MLA's namely Random Forest and achieved accuracy of 78%. The steps followed in applying Machine Learning algorithm (Random forest) on the text data are: 1. Created the corpus, 2. Clean the Text data in corpus using tm_map(), 3. Removing all the English stop words, 4. perform stemming on the corpus, 5. creation of Document Term Matrix (DTM), 6. partitioning the data as training (75%) and testing (25%) datasets. 7. Apply Random forest algorithm to obtain confusion matrix. 8. Obtained the Accuracy of the model from confusion matrix as shown in the Eq. (1)

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN} \tag{1}$$

Where TP is Total number of True Positives, TN is True Negative, FP is False Positive and FN is False negatives. The values obtained in the experiment is plotted in the Eq. (2)

$$Accuracy = \frac{12 + 105}{(12 + 105) + (28 + 5)} = 78\% \tag{2}$$

Similarly the other models like Support Vector Machine (SVM), Navie Bayes (NB) and Decision Tree (DT) can be applied to the text and achieve better accuracy.

The research work carried out by the author in [9], is on classification on teacher rating in kinder gartners for language have achieved 77% of sensitivity. Whereas, our approach yields to have 78% of classification accuracy on the dataset considered in the experiment.

IV. CONCLUTION

In this paper, our aim is to covert the input Text data in to image format and apply MLA on the data and predict the accuracy. In the experiment conducted, Random forest algorithm is applied and achieved 78% of accuracy. Towards presenting the text data in to image, that helps in deriving insights from Text data, the same is plotted in methodology section. The limitation of our study is only random forest is applied and achieved the 78% accuracy. Yet to apply other Machine Learning algorithms and compare the performance in terms of accuracy. In Future, we would like to list out the advantages and drawbacks of each techniques in representing the text as image and apply different machine Learning algorithm

References

1. Basha, Syed Muzamil, and Dharmendra Singh Rajput. "**Sentiment analysis: using artificial neural fuzzy inference system**." *Handbook of Research on Pattern Engineering System Development for Big Data Analytics*. IGI Global, 2018. 130-152.
2. Basha, S. M., & Rajput, D. S. (2018). **Parsing Based Sarcasm Detection from Literal Language in Tweets**. *Recent Patents on Computer Science*, *11*(1), 62-69.
3. Basha, S. M., & Rajput, D. S. (2018). **A supervised aspect level sentiment model to predict overall sentiment on tweeter documents**. *International Journal of Metadata, Semantics and Ontologies*, *13*(1), 33-41.
4. Basha, S. M., & Rajput, D. S. (2017, December**). Evaluating the impact of feature selection on overall performance of sentiment analysis**. In *Proceedings of the 2017 International Conference on Information Technology* (pp. 96-102). ACM.
5. Belford, M., Mac Namee, B., & Greene, D. (2018). **Stability of topic modeling via matrix factorization**. *Expert Systems with Applications*, *91*, 159-169.
6. Zhu, M., Xia, J., Jin, X., Yan, M., Cai, G., Yan, J., & Ning, G. (2018). **Class weights random forest algorithm for processing class imbalanced medical data**. *IEEE Access*, *6*, 4641-4652.
7. Şenel, L. K., Utlu, I., Yücesoy, V., Koc, A., & Cukur, T. (2018). **Semantic structure and interpretability of word embeddings**. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(10), 1769-1779.
8. Kusumaningrum, R., & Adhy, S. (2018). **WCLOUDVIZ: Word Cloud Visualization of Indonesian News Articles Classification Based on Latent Dirichlet Allocation**. *Telkomnika*, *16*(4).
9. Gregory, K. D., & Oetting, J. B. (2018). **Classification accuracy of teacher ratings when screening nonmainstream English-speaking kindergartners for language impairment in the rural South**. *Language, speech, and hearing services in schools*, *49*(2), 218-231.