A Project Report

On

Word Embedding in Text Classification

BY

SHIVANG SINGH

2018A7PS0115H

Under the supervision of

Dr. LOV KUMAR

SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS OF

CS F266: STUDY PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

HYDERABAD CAMPUS

(October 2020)

ACKNOWLEDGMENT

This project wouldn’t have been possible without the support from Bits Pilani and I am very grateful to them for granting me with this opportunity.

I would like to express my sincerest gratitude to my mentor, Dr. Lov Kumar, for his guided support throughout the course of the project. I would like to thank my Mentor for providing me with such a wonderful opportunity to work on real life data and get hands on experience on such a fast-growing topic.

I am also grateful to my colleagues and friends who have helped me in making great progress throughout the project. I would also like to thank my parents for their personal support and motivation which has helped me to continue all the way.

Birla Institute of Technology and Science-Pilani,

Hyderabad Campus

Certificate

This is to certify that the project report entitled "Word Embedding in Text Classification" submitted by Mr. SHIVANG SINGH (ID No. 2018A7PS0115H) in partial fulfillment of the requirements of the course CS F266, Study Project Course, embodies the work done by him under my supervision and guidance.

Date: 23-10-2020 **(Dr. Lov Kumar)**

BITS- Pilani, Hyderabad Campus

ABSTRACT

Due to an increased demand for an accurate vectored representation of textual data in the field of Natural Language Processing, Word Embedding has become more relevant. Important research areas such as Speech Recognition, Text Analysis, Feedback Forums, etc indicate the need for a structured approach to handle and prepare textual data for Machine Learning Algorithms.

This Project broadly deals with the "Classification of Feedback Tweets for various Airlines based on their Sentiment". The aim is to classify tweets into 3 Categories namely positive, negative or neutral based on the textual information provided through the tweet.

The Project focuses on recognizing the most suitable Word Embedding Technique, Approach to handle Imbalanced Data, Feature Selection and Classification Technique for the given data set to arrive at the most suitable Framework for the solving the problem.

TABLE OF CONTENTS

FRAMEWORK

1. The Dataset consists of the Classification label (airline_sentiment) and the Text field(text), which specifies the body of the tweet.

2. Apply various **Word Embedding Techniques** such as glove, W2V, CountVectorizer, TFIDF, skip-gram, fasttext, bert, gptmodel, etc to the text field of the dataset.

3. Identifying the Imbalance in the dataset and handling it using various **SMOTE** (Synthetic Minority Over-Sampling Technique) techniques such as SVSMOTE, SMOTE, Borderline SMOTE, SMOTENC.

4. Selecting appropriate features using various **Feature Selection Techniques** such as Correlation analysis, Principle Component Analysis, Significant Features.

5. Applying various **Deep Learning Algorithms** and analyzing their performances to gauge the best combination of Word Embedding, SMOTE Technique, Feature Selection Technique and Deep Learning Algorithm for the given dataset.

DATASET DESCRIPTION

- The Dataset has been obtained through kaggle titled “Twitter US Airline Sentiment”. It provides information about how travelers in February 2015 expressed their sentiments on Twitter. The aim is to analyse the sentiments of a tweet based on its contents.

Twitter US Airline Sentiment

- The Classification label used here is the ‘airline_sentiment’ label. It takes 3 categories namely, Positive, Negative, Neutral.
- The Text field of interest is the ‘text’ label, which contains the actual contents of the tweets provided by the travelers.
- The dataset consists of 14640 data entries and the distribution with respect to the classification label is as follows:

Classification Class	Distribution (%)
Positive	16
Negative	63
Neutral	21

- The distribution indicates imbalance of data among the classification categories and hence indicates the requirement of various SMOTE Techniques.
- The dataset also provides other useful information such as reasons for negative and positive sentiments thereby providing useful feedback to major US Airlines.

WORD EMBEDDING TECHNIQUES

- Word Embedding refers to a set of modelling techniques that are used to provide contextual meaning to Textual information by mapping words or phrases to a learned vector of real numbers.
- They are important as they provide useful mathematical correlations among similar words and act as appropriate inputs for various Deep Learning Algorithms.
- It allows the user to capture the syntactic and semantic meaning of textual information in a format that can be provided as input to Deep Learning models. It improves the performance of text classifiers.
- In this Project, different Word Embedding Techniques are applied to the 'text' labeled field to provide various possible vectored representations of the same dataset.
- The Word Embedding Techniques applied to the dataset are as follows:
  - TFIDF (Term Frequency-Inverse Document Frequency)
  - Count Bag of Words
  - W2V (Word to Vector)
  - GloVe (Global Vectors for Word Representations)
  - Skip-Gram Model
  - FastText
  - BERT Model
  - GPT Model (Generative Pre-Trained Transformer)

- **TFIDF (Term Frequency- Inverse Document Frequency)**

  This method specifies the vocabulary of the Word Embedding using the input data. Most common words (specified in terms of minimum occurrences within the dataset) are used as column representations and their normalized frequencies within each document are captured. It captures no contextual meaning and correlation among words. It generates a sparse matrix and is usually used to compare documents. The dataset generated is sparse and large and hence is not used as a Word Embedding Technique since we also apply SMOTE oversampling to handle imbalanced data.

- **Count Bag of Words (CBOW)**

  This model allows us to provide context to words by targeting its surroundings. It uses surrounding words (context) to predict the current word. It requires a context-window length which specifies how far from a given word does the relevant context extend. It is an implementation approach to the Word2Vec model but in this case we use our own dataset to train the model.

- **Skip-Gram (Skg)**

  Similar to the Count Bag of Words model, even this model uses surrounding words as context for a target word but it is different in the sense that it works in an opposite manner. It takes a target word as input and predicts using various probability distributions the most likely context window for it. It is also an implementation approach to the Word2Vec model. In this case as well, we use our dataset to train the model. It works on a semi-supervised learning based approach.

- **GloVe**

  It is an Unsupervised learning algorithm, that is used to create global representation of words. The model for our dataset is trained using glove 6B 300d dictionary. The model provides us with a [300X1] vectored representation for each word in our document. We can compute the semantic similarity between words using the Euclidean distance between two vector representations. It's drawback lies with the fact that there are only few relationships between words that can be captured using a single number. The [300X1] representation for a sentence has been obtained by averaging the [300X1] representations for all its constituent words.

- **FastText**

  It is an extension of Word2Vec model. Instead of having a vectored representation for a word, it produces n-gram of characters for each word. This consequently captures the meanings of prefixes and suffixes. It is especially useful for getting the embedding for words that have not been seen before while training the model since it can be broken down and analyzed. The data corpus used to train the model is the English Wikipedia .

- **Word2Vec**

  It is a basic implementation of Word2Vec model which maps a given word to a learned vector. We have already seen its two implementation approaches namely, Count bag of words and Skip-gram model, but in both those approaches the model was trained using the dataset. In this case, we use an existing data corpus provided by Google called Google News Vector negative 300.

- **BERT Model**

It is based on a transformer architecture rather than the conventional LSTM (Long Short Term Memory) architecture. It stands for Bidirectional Encoder Representations from Transformers. It is trained on a very large data corpus of unlabeled text including the entire Wikipedia Book Corpus. This large corpus allows it to deduce deeper meaning and relations between words. The bidirectional aspect of it comes from taking into account the left and the right context for a word. Once the model has been trained using the corpus it is fine-tuned to specific NLP tasks. It is pre-trained on two tasks namely, Masked Language Modelling and Next Sentence Prediction.

- **GPT Model**

It is also based on a transformer architecture. It works on the principle of using pre-trained models to create the corresponding word embedding for our dataset. It finds its application in Text Auto-correct, Auto-complete feature on IDE's. Unlike BERT, it has decoder blocks instead of encoder blocks. It is an autoregressive (predict future behaviour based on past behaviour) learning approach that produces human-like text. It is proven to be extremely powerful at Summarizing Texts and Answering Questions.

IMBALANCED DATA TREATMENT

- As was observed while analyzing the dataset, the distribution of the classification parameter (airline_sentiment) was found to be skewed/biased.

- Since most machine learning algorithms operate on the basic assumption that there are equal number of examples for each classification category, balancing the data becomes really important and it also increases the predictive power of the model.

- The source of such imbalance could be attributed to various factors such as biased sampling, measurement errors, problem domain, etc.

- One way to treat imbalanced data is by over sampling the minority class using existing data entries. This method is known as SMOTE (Synthetic Minority Oversampling Technique). It generates new data entries for minority class to balance the training examples for each class.

- For each Word Embedding obtained through various techniques for the given dataset, we apply various SMOTE Techniques to generate various balanced datasets depicting all combinations of Word Embedding Techniques and SMOTE Techniques.

- The various types of SMOTE Techniques applied to the dataset are:
  - SMOTE
  - SVSMOTE
  - Borderline-SMOTE,
  - SMOTE-NC,
  - ADASYN.

SMOTE TECHNIQUES

- **SMOTE**

  It is the most default form of SMOTE Technique which just over samples the minority class. It works only for continuous data features.

- **SVMSMOTE**

  It is an extension to borderline SMOTE but it uses the SVM algorithm to identify misclassifications. It focuses more on the parts of the distribution where data is separated rather than generating data on minority class overlap.

- **Borderline-SMOTE**

  It is similar to SMOTE except for the fact that it generates data only along the decision boundary between the two classes whereas SMOTE generates data randomly between two data points.
  It has two variations :
  1. Borderline SMOTE1 - Oversamples the majority class as well in case of misclassifications along the decision boundary.
  2. Borderline SMOTE2 - Just oversamples the minority class.

- **SMOTE-NC**

  It is just an extension to the default SMOTE Technique but it works for categorical and continuous features as well. It is especially useful for cases when we have mixed features in our dataset.

- **ADASYN**

  It stands for Adaptive Synthetic Sampling. It is similar to borderline SMOTE except for the fact that it generates the data according to data density. Data generation is inversely proportional to the density distribution of the minority class.

FEATURE SELECTION

- After removing imbalance from the data, we are now left with multiple datasets having numerous features (column labels) representing each document in our dataset.

- An important step to be completed before the dataset is ready to be given as an input to a Deep Learning model is Feature Selection.

- Feature Selection refers to the set of methods/techniques that are used to select a subset of input features by eliminating redundant, irrelevant and non-informative features from the dataset. It improves the predictive accuracy of the DL models.

- It reduces the size of the dataset and hence allows the model to train faster. Often the features are ranked based on their significance in predicting the classification class and only the top log n features are chosen (varies based on the total features).

- The need for applying Feature Selection arises from the presence of Correlated input features which reduce the relative importance of the actual significant features. Some of the features are relatively more important than the others in identifying the class for a data entry. Usually, the features are ranked based on certain specific property which is specified through Domain knowledge and Requirements.

- The Feature Selection Techniques used for the datasets are as follows:
    1. Significant Features
        i. Gini Values
        ii. Confidence Intervals
    2. Correlation Analysis
    3. Principal Component Analysis (PCA)

- **SIGNIFICANT FEATURES**

  - This method essentially depicts the class of techniques that are used to the determine the most relevant/significant subset of the set of input features which are useful to determine the class of a given input. It essentially involves different properties based on which the features are ranked for significance.

  -We use two common implementations of this approach:

  1. **Confidence Interval -** Out of all the features in the dataset, only those features are selected for which the 95% Confidence Intervals for different class categories do not overlap. CAn be visualized more easily using Box plots.
  2. **Gini Index -** It calculates the probability that a specific feature is classified incorrectly when chosen randomly. It works for discrete values only.

- **CORRELATION ANALYSIS**

  -This method is used to remove correlated input features from the dataset. For our dataset, we have used **Pearson** Correlation as a measure for correlation.

  -For each input feature, Pearson Correlation returns a value between -1 and 1. Usually, an absolute value of >=0.7 is treated to be highly correlated and hence one of those two features is dropped from the dataset.

  -The above method is applied for each pair of input features until only uncorrelated features are left. This can also be visualized using correlation heatmaps (present in seaborn library)

- **PRINCIPAL COMPONENT ANALYSIS (PCA)**

  -It is not essentially a feature selection technique but can be used as one if the property that depicts the significance of an input feature based on its variation.

  -It creates new uncorrelated variables that can be expressed as linear combination of old variables thereby minimizing the size of the input features without having drastic effects on the information conveyed.

  -It is one of the most common dimensionality-reduction techniques.

REFERENCES

- CountVectorizer
- TFIDF
- Word2Vec
- GloVe NLP , GloVe Embeddings
- Cbow vs Skip-Gram Model
- FastText
- LSTM vs Transformer Architecture
- GPT Model
- BERT Model
- Identifying Data Imbalance
- SMOTE Techniques
- Feature Selection Importance
- Feature Ranking and Feature Subset Selection
- Confidence Interval and Pearson Correlation
- Principal Component Analysis