Ternary Classification technique based on Factorization

D.Dineshwar, (3rd Year E.C.E, C.M.R.I.T, dineshsupertechno@gmail.com)

G.Yeshwanth (3rd Year E.C.E, M.R.C.E.T,yeshu.mrcet2019@gmail.com)

Abstract: The main aim of this project is to develop a classical method for classifying 3 tasks based on factorization. Today even we had betters ways to classify like using neural networks which requires heavy computational power and large datasets; we can use this technique to classify 3 tasks with medium size dataset and requires relatively low computational power. This technique is developed to use in field of Natural Language Processing (N.L.P).This method can be very useful if you have relatively low dataset and low computational power. This method does not even require Tokenization and Stemming, so this makes this method relatively simple.

Introduction: Classification is supervised machine learning technique which is used to identify the given input belongs to which category of data. The main aim for classification is to learn how to classify the given input data. Unlike regression technique the output of Classifier is always a discrete number which represents the class to which it belongs. Here a ternary classifier is a classification technique which predicts the most suitable class for given input. Here this technique is based on Factors of prime numbers. This technique eliminates the process of stemming and tokenization by allotting prime numbers multiples to data and eliminating the common words which are between the classification groups. This makes this method to consume low amount of computational power and requires low or medium sized datasets. This whole method is based on assigning prime numbers multiples to each example in dataset and eliminates the common words by looking at the common numbers among examples across dataset. This process requires only removing of keywords from given dataset.

Baseline approaches: Actually there are pretty good ways to solve this problem but always there is a trade-off between amount of data required and accuracy, as the amount of data increases time required to compute increases which forces to increase computational power in order to save time. For better accuracy people use LSTM's(Long Short Term Memory) cells of RNN's(Recurrent Neural Networks) but for using these methods we require heavy datasets in order of million. But if our main concern is short data then we go with Classical methods which may not have high accuracy as LSTM's but it still works on low dataset requirements one of such methods is using frequency distribution among examples of dataset and as dataset size increases accuracy increases at a point it reaches threshold where it can't achieve more accuracy. So both of these are on extreme ends of tradeoff , where in one accuracy is high while in other small datasets are even acceptable.

Proposed approach: The method which we present in this paper works on relatively small datasets and acquires the threshold accuracy of Classical frequency methods.

ALGORITHM (building predictive weights):

1. **Remove all stop words from the given dataset.**
2. **Now assign numbers 2, 3, 7 to classes in the given dataset.**
3. **Now multiply each and every example with even numbers starting from 2 to 'n' (i.e. n is number of example belongs to a single class in given dataset)**
4. **After multiplying with even numbers Now make a dictionary that contains all words which are keys and their values are the count of number associated to their example in given dataset and add them among all the classes in dataset.**
5. **Now check each and every key if it's values is a multiple of number's 2, 3 and 7. If it's a multiple of all three numbers then delete the key value pair from that dictionary**
6. **Running this process isolates the word which mainly constitutes the meaning of the task and removes the rest of**

ALGORITHM (Computing the output):

1. **Take the input and create 3 variables each one for each class.**
2. **Now add the values to variables which belong to particular class by looking at dictionary. Continue all this process for all words in given input.**
3. **Now find the maximum variable of all three variables.**
4. **Depending on maximum variable output the class belongs to respective class.**

Result: The above method proposed works relatively better on low datasets. This makes this method better than on other methods because this is in between the ends of dataset size and accuracy scale.