4222 – SURYA GROUP OF INSTITUTIONS

VIKRAVANDI

SENTIMENT ANALYSIS FOR MARKETING

Prepared by,

K.SURESH

422221106016

ECE-III Year

SENTIMENT ANALYSIS FOR MARKETING

AI _Phase 2

MACHINE LEARNING

INTRODUCTION:

One application of machine learning is in sentiment analysis. In this field, computer programs attempt to predict the emotional content or opinions of a collection of articles. This becomes useful for organizing data, such as finding positive and negative reviews while diminishing the need for human effort to classify the information.

Machine Learning approach :

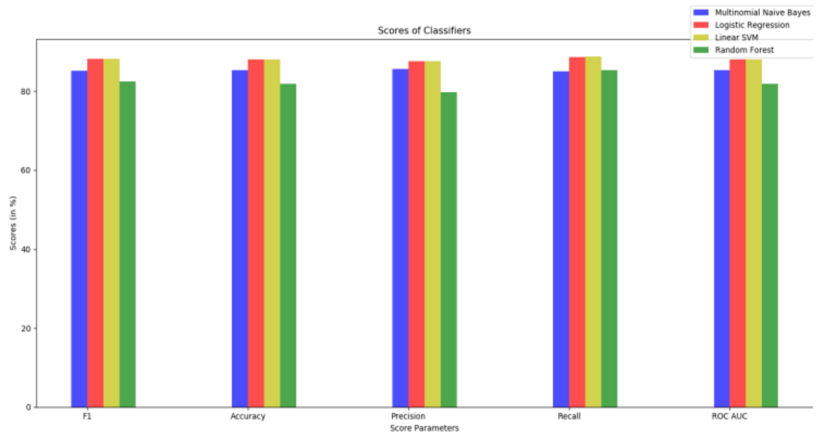
Machine Learning approach is widely seen in the literature on sentiment analysis. Using this approach the words in the sentence are considered in form of vectors, and analyzed using different machine learning algorithms like Naïve Bayes, SVM, and Maximum Entropy. The data is trained accordingly, which can be applied to machine learning algorithms. The detailed discussion on Machine learning approach is discussed in Chapter.

Data Set and Variables :

The Twitter Data available is of World Cup Brazil 2014 '#brazil2014','#worldcup2014', and games hashtags, as shown in Table 2. This data set distinguish the tweets based on the hashtags namely #brazil2014, #worldcup2014, #ALGRUS (Algeria vs Russia) as well as, other games and event. The hashtag #worldcup2014 contains all the tweets from the date 06-June to 14-July, 2014 (40 Days), which consist of 44,040,192 user tweets globally during the world cup. Similarly, the hashtag #brazil2014 comprises of all the user tweets on the promotion of the world cup.

Comparison to Human Prediction:

One might ask what is the difficulty of our two tasks and what level of accuracy would be considered successful. To answer the question of how hard the two tasks are, we can compare our system's performance against that of humans. We conducted an scaled-down version of the experiment where we had humans attempt the same two classification task as our models. Performance at the human level is often considered the target goal in sentiment analysis.



we are not asking humans what are their reactions; we are asking them to predict what they think .

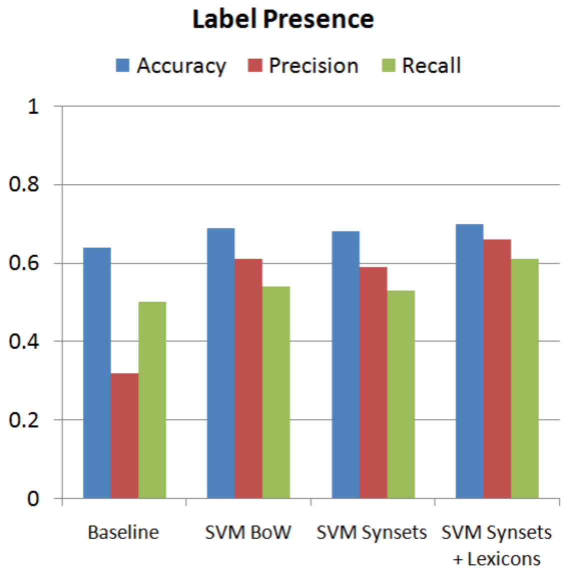


Figure 3: Performance of different models on label presence task.

Other Attempt:

In addition to what we used in our final model, we had other work that taught us more about extracting emotion from EP. 3 For the max label task, due to the unbalanced distribution of categories we used a balanced human testing set instead of a random subset of the original testing set. Note that this is a harder problem for our SVM classifier since it was trained on an unbalanced training set. As a result the numbers reported here are lower than the ones reported in Results. Figure 4: Human comparison for max label.

```
params = list(model.named_parameters())

print("The BERT model has {:} different named parameters.".format(len(params)))

print("==== Embedding Layer ====")
```

```
for p in params[0:5]:
  print("".format(p[0], str(tuple(p[1].size()))))

print("==== First Transformers ====")
for p in params[5:21]:
  print("{:<60} {:>12}".format(p[0], str(tuple(p[1].size()))))

print("==== Output Layer ====")
for p in params[-4:]:
  print("{:<60} {:>12}".format(p[0], str(tuple(p[1].size()))))
```

Input: POS (Part-of-speech) tagged word, negation marks ('1' for Negative or '0' for Positive)

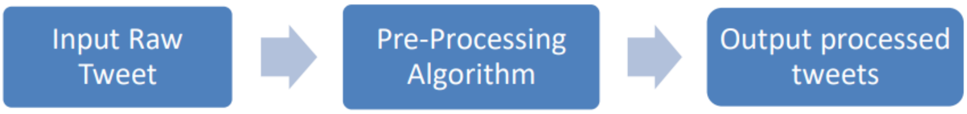
Output: A unique synset word with its part of speech and close meaning to the word.

Method GetSynset by passing POS tag word and Negation mark

    Method to Sanitize part-of-speech (POS) tag to WordNet accepted POS
    For **synset** in WordNet Synsets (word, POS tag):
        Returns list of synsets for the words
        For **lemma** in synset list:
            If word equals to lemma name
                Append **Synonyms**(word with the same meaning) list
            If word has its Antonyms
                Append **Antonyms**(word with opposite meaning) list
    If negation mark is **'0'** and it is not NULL
        Return first **synonym** of word and POS tag from Synonyms list
    Else
        Return the same word and POS requested
    Else IF negation mark is **'1'** and it is not NULL
        Return first **antonyms** of word and POS tag from Synonyms list
    Else
        Return the same word and POS requested

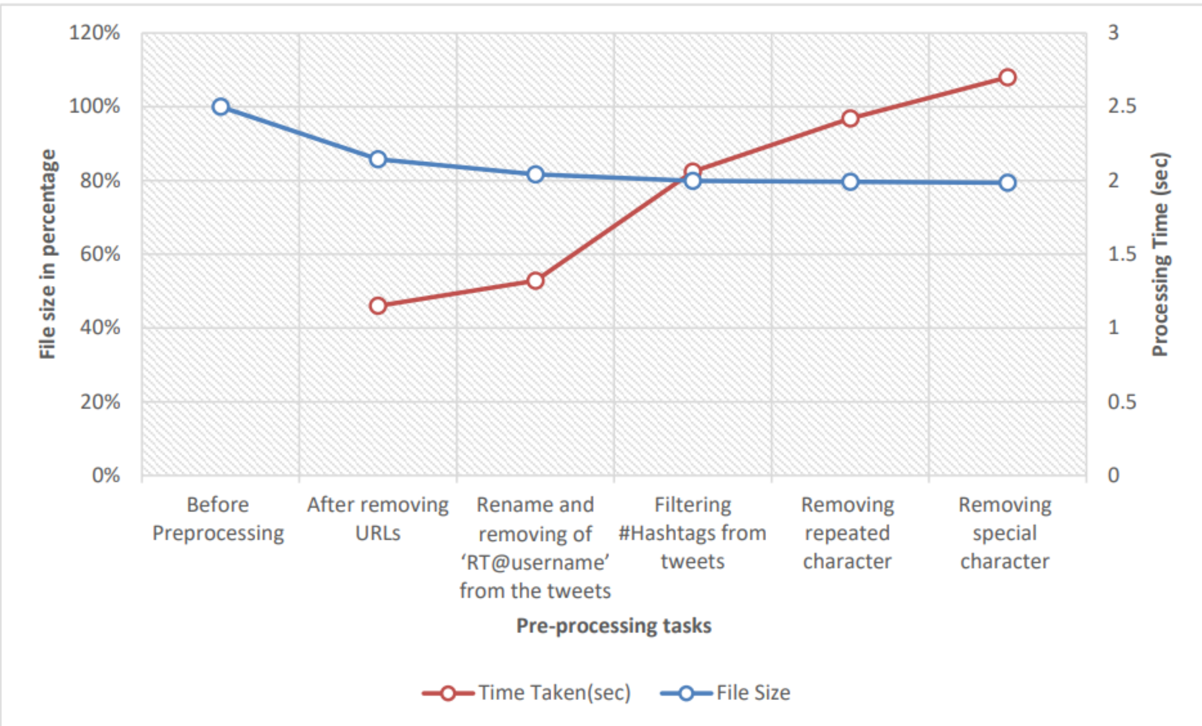
Data Cleaning and Noise Reduction:

Data set available on world cup 2014 contains text field, in which user's comments or tweets information on particular event or game is available. These tweets are in unstructured form



Furthermore, the user generated information may also contain unnecessary whitespaces at the beginning, in between or at the end of the tweets, special characters like punctuation and repetition of characters. First, all extra white space was removed using the build in function available in Python. Secondly, all the meaningless and unnecessary special characters from the tweets were eliminated (Hemalatha et al. 2012). These characters include: \ | [ ] ;: {} - + ( ) < > ? ! @ # % *, and a few more. Neither do these characters have specific and special meaning, nor do they explain if these characters are used for positivity or negativity, hence; removing them is the best option

- Non-standard (slang) to standard word mapping
- PoS tagging
- Tagging Stopword removal
- positive/negative words

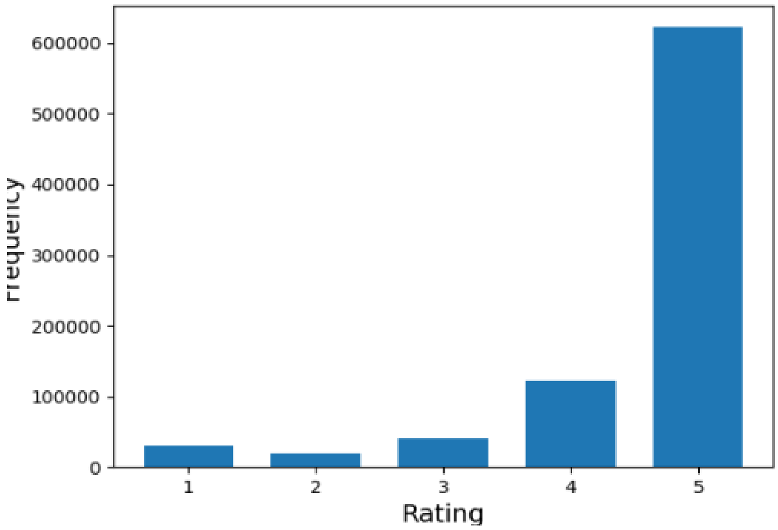


```
import numpy as np

print('---Train---')
print('input: ', train_input.shape)
print('label: ', train_labels.shape)
print('mask: ', np.array(train_mask).shape)

print('---Validation---')
print('input: ', validation_input.shape)
print('label: ', validation_labels.shape)
print('mask: ', np.array(validation_mask).shape)

print('---Test---')
print('input: ', test_input.shape)
```



CONCLUSION:

Sentiment analysis deals with the classification of texts based on the sentiments they contain. This article focuses on a typical sentiment analysis model consisting of three core steps, namely data preparation, review analysis and sentiment classification, and describes representative techniques involved in those steps.