CMPUT 501 Project Proposal- SemEval 2020 Task 8 : Memotion Analysis

Shruthi Ravishankar and **Afia Anjum**
Department of Computing Science
University of Alberta

Abstract

Memotion (Meme + Emotion) Analysis is the task of analyzing sentiments from memes, typically posted in various social platforms, and classifying them into positive and negative sentiments. We present a methodology that represents our utilized approach to extract and analyze embedded texts of a meme from a released data set (training and trial) of a SemEval-2020 task titled as ”Memotion Analysis”. We utilize a Naive Bayes classifier to classify our texts into sentiments, and logistic regression classifier is used as a baseline approach to compare the results of the classification. We have measured F1 score for both the approach and have obtained a score of 93.00% and 90.86% for the Naive Bayes classifier and Logistic Regression classifier respectively.

1 INTRODUCTION

With the abundance of memes in the social media such as Facebook, Reddit, Twitter etc, extraction of the real meaning behind these memes can be important for a couple of reasons. One is that it would help us identify the emotions of the users in that social platform as well as identify the community that this meme is targeted towards. It has been observed that, sometimes a meme would cross the demarcation line of being sarcastic and turn itself into a negative post hurting a specific person or a community. Besides, a meme would also represent social and cultural experiences of a community, which in turn will help us in identifying the opinion that a community holds on certain matters.

Many social media companies rely on hired people to remove offensive contents and memes from the stream data. Such manual intervention is very inefficient and expensive to perform. Thus, analyzing positive and negative sentiments in memes is very important in the present day. So far, very little research has been done in this domain leveraging Natural Language Processing tools and techniques. Hence, our goal is to contribute more to this hybrid approach of analyzing embedded texts from memes and then classifying their inherent sentiments into two categories. The input to the system would be an embedded text from a meme and the output would be the sentiment of the meme.

The following section presents the works that are highly or partially related to our task. This section is followed by a Methodology section which introduces our utilized approach in detail. Section 3 discusses about the metrics used to evaluate our model, which is followed by a system architecture and a pseudocode in section 4 and 5 respectively. The final two sections of this paper deals with the results obtained from our selected model and concludes the future aspects of this task.

2 RELATED WORK

Some researchers have studied the sentiment analysis of images while others have worked on the sentiment analysis of text. The paper titled as Emotion Detection and Sentiment Analysis of Images (Gajarla and Gupta, 2015) uses the Flickr dataset to train neural network models in order to classify images into different emotion categories including violence, love, happiness, sadness and fear. However , the text embedded in memes plays a crucial role in determining the sentiment of the meme. The above described method is done using only image processing, and the text embedded within the images are ignored. We take into account the text of the meme to categorize them as

having positive or negative sentiments.
On the other hand, another work titled as Identifying Expressions of Emotion in Text (Aman and Szpakowicz, 2007) describes a task that involves annotation of emotions from text into different categories. In this research, data is extracted from a variety of blogs including news articles and diaries and then trained using Naive Bayes and Support Vector Machines classifiers. However, as mentioned above, both the content of the image as well as the text embedded within, is needed to classify the image into the positive or the negative sentiment categories and this method fails to achieve our goal of classifying memes.
Observing all these research, we can conclude that, there has been no work done so far utilizing knowledge of both text and image together and classifying the memes accordingly into positive and negative sentiments. We focus on the same, using the information from both the images and the text collectively, to perform sentiment analysis on memes.

3 METHODS

The following section describes the methods used to build the system as well as the steps taken to improve the predictions.

3.1 DATASET

The dataset is obtained from Sem-Eval 2020. The training data is a comma separated value file that consists of 6601 rows of data with nine features, which are as follows :

- Image name
- Image URL
- OCR Extracted Text
- Corrected text
- Extent of humour
- Extent of sarcasm
- Extent of offensiveness
- Extent of motivation
- Overall sentiment

Since the final goal of our system is to classify the meme into a positive or a negative sentiment meme, hence, we make use of only two features from the training data : OCR Extracted text and the Overall Sentiment.

3.2 PRE-PROCESSING

The original dataset consists of nine different features, out of which we will consider two important features, the OCR Extracted Text, which is the text embedded within the meme, as well as the overall sentiment of the meme. The overall sentiment is further split into five categories :

- Very positive
- Positive
- Very negative
- Negative
- Neutral

The two features , as mentioned above are first extracted from the dataset, and pre-processed in such a way that the extracted text corresponding to the first two sentiments (very positive and positive) are appended to a positive list and the extracted text corresponding to the next two sentiments (very negative and negative) are appended to a negative list. The extracted text containing neutral sentiment is ignored, for, it isn't related to the purpose of our system.

We make sure to remove any punctuation from extracted texts in the two lists so that the nltk tokenizer can accept plain texts as input and the Naive Bayes classifier can deal with words from this tokenization as features. We convert all the words in each of these texts into lowercase letters to prevent counting uppercase and lowercase instances of the same word separately. The final cleaned dataset results in a list containing 3771 positive items and in another list that contains 563 negative items.

3.3 K-FOLD CROSS VALIDATION

K-fold cross validation is used for selecting the best parameters, so that we get the highest accuracy possible. The parameters considered here are the ”removal of stop-words” and the ”removal of unknown words”. K-fold cross validation is performed here by splitting the training data into three splits,training on two splits and validating on the third split. The best parameters are the ones that produce the highest accuracy and these parameters are used to train the Naive Bayes classifier.

3.4 BASELINE SYSTEM

A baseline result is the simplest possible prediction. For some problems, this may be a random result, and in others in may be the most common prediction. Logistic regression is used as a baseline classifier for this system. Since, Naive Bayes takes the class prior probabilities into consideration, our hypothesis is that the Naive Bayes classifier, being modelled on this particular data set, would give us a better performance result than the Logistic Regression.

3.5 TRAINING AND TESTING

The pre-processed data is trained using the Naive Bayes classifier, which calculates the prior probabilities, P(c), and the likelihood, P(W|c)), of each word belonging to a particular sentiment (either positive or negative). The system is tested using the test data acquired from Sem-Eval 2020 and the evaluation metrics are reported. The final posterior probability for each word is the product of the prior probability and the likelihood of the word belonging to a particular sentiment. The formula is summarized below.

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} \log P(c) + \sum_{i \in positions} \log P(w_i|c)$$

3.6 EVALUATION METRICS

The system is evaluated using precision, recall and F1 score. Additionally, a confusion matrix is drawn for determining the true and false positives (TP and FP) and true and false negatives (TN and FN).

The following part discusses each of this metrics in detail

- **Accuracy** The accuracy of the Naive Bayes classifier on the test data is calculated taking each of the hyper-parameters into account during the cross validation phase only. However, to measure the performance of our classifiers, we ignored the accuracy score because of it's inherent misleading nature. Since we are dealing with an imbalanced class data set, it's the best approach to avoid looking at the accuracy score.
- **Precision** Precision attempts to answer the following question: What proportion of positive identifications was actually correct? Precision is defined as follows :

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall** Recall attempts to answer the following question: What proportion of actual positives was identified correctly? Recall is defined as

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1 scores** In binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. The F1 score is the harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$F_1 = \frac{2PR}{P + R}$$

- **Confusion Matrix** In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.
- **Receiver Operational Characteristics (ROC) Curve** ROC curve is a another performance measurement for classification problem. Generally this curve is used to visualize the true positive rates (TPR) and false positive rates (FPR), obtained from a classification model, in a better way. By achieving a high TP rate and a low FP rate, one can be confident about the performance of a model.

4 SYSTEM ARCHITECTURE

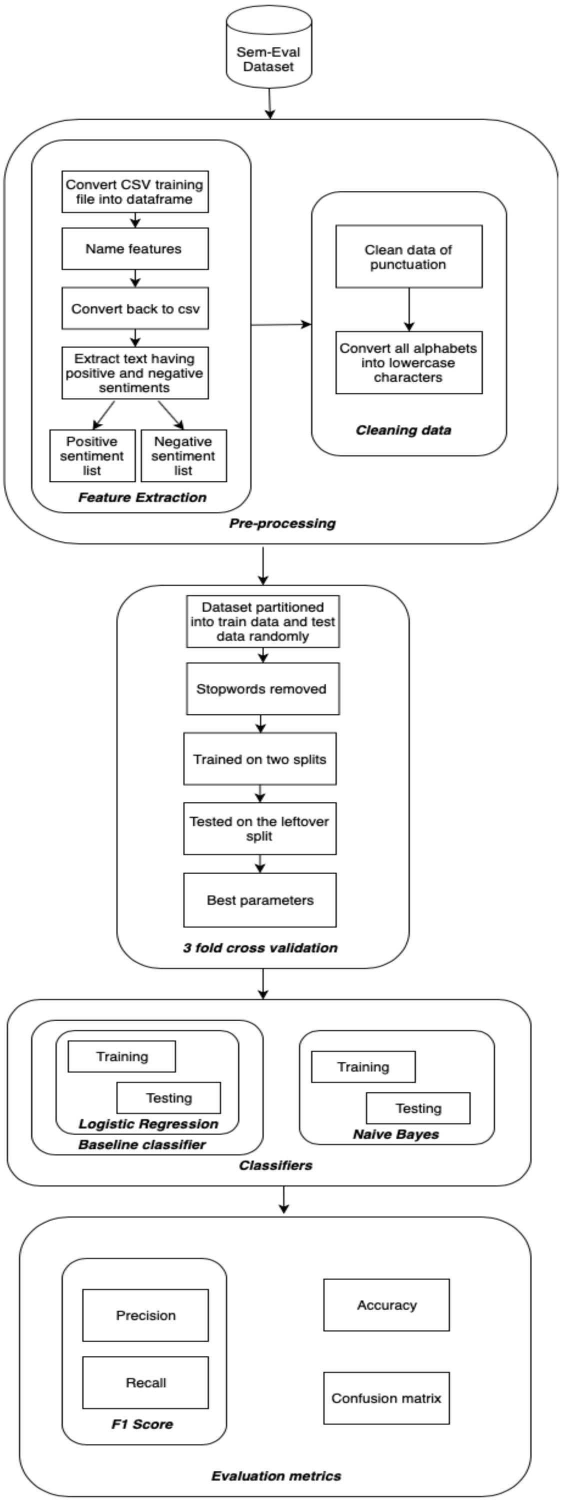

Figure 1: System Architecture Diagram

As shown in the system architecture diagram (Figure 1) , the training data is first filtered in such a way that the data from the comma separated value file is split into two columns, one for positive and one for the negative sentiment sentences.The data is cleaned and the features from the training file are extracted and passed to the Naive Bayes classifier. The reason that this particular classifier is used, is that the Naive Bayes classifier ignores the irrelevant features and can perform well even if some of the data is missing. The model is evaluated using precision, recall and the F1 score of the classifier. A confusion matrix is also drawn that can be used to describe the performance of the system.

5 PSEUDO-CODE

```
Algorithm 1: Memotion Analysis
 Input : A CSV file that contains the
  extracted text from memes along with the
  sentiment classification.
  Output : Positive or Negative Sentiment.
  1) Create a python dataframe that contains
  positive and negative sentiments along
  with the extracted texts.
  2) Feature Extraction; Extract ’positive’
  and ’negative’ features from the python
  dataframe.
  3) Apply a K fold cross validation on the
  given training set that in turn would return
  the best parameters.
  4) Use the selected best parameters to train
  the Naive Bayes Classifier.
  5) Test the classifier with the given test set
  and outputs the precision, recall and F1
  scores of classification.
```

6 RESULTS

The model is tested and the results are recorded, as shown below.

- **Baseline Classifier-Logistic Regression:** A logistic regressor is chosen as the baseline system, where the best step-size is selected to be 0.001 with the help of cross validation. The model is tuned with respect to following values of step sizes such as, 0.001, 0.01, 0.1 and 1. The F1 score obtained for our logistic regression classifier is around 90.86 percent, which is shown below.

Metrics	Scores
Avg. accuracy with CV=3	88.83%
Precision	0.88
Recall	1.0
F1 Score	90.86%

From the above table it can be seen that the logistic regression achieves a precision score of 0.88 and a high recall of 1.0. Though it seems like a reasonable performance, looking at the entries of the confusion matrix below clears our suspicion. The classifier does a very good job of predicting an originally positive meme as positive and fails in detecting any of the original negative meme as negative. Other drawback of this classifier is that it also identifies a large portion (69) of negative test examples as positive.

Labels	Actual Pos	Actual Neg
Sys. Pos	549	69
Sys. Neg	0	0

- **K-fold cross validation:** The value of K is taken as 3, which means the training data set is split into three parts and cross validation is performed by training on two parts and testing on the third part. The hyper-parameters considered here are the stop-words and unknown words, and we analyse how the removal of them affects the result of the cross validation. As shown in the Figure 3, we can see that the best hyper-parameters are the ones where we remove the stop-words and we ignore the unknown words as and when they occur in the test data. It is shown in the following table.

Parameters	CV Accuracy
Rem. Unk. & St. words	92.12
Keep Unk. & St. words	91.02
Keep Unk. & rem. St. words	92.12
Rem. Unk. & keep St. words	91.02

- **Accuracy** The accuracy of the Naive Bayes classifier on the test data is calculated. However, we get an accuracy of around 89 percent. However, we will ignore this metric for this particular classification task. This is because, an imbalance is observed in the training data, where we have 3771 positive sentiment data and 563 negative sentiment data. This imbalance would mislead our classifier. Thus, we take a look at the F1 scores of the classification, to have a better idea of how our model works.
- **F1 scores** The F1 score is 0.93 for our Naive Bayes Classifier, while the precision is 0.92. It means that around 92 percent of our identifications were actually correct. The recall is at 0.95 which is the proportion of our samples that were identified correctly. It is to be mentioned that, while computing F1 score, we should be more concerned about precision than the recall for the type of our classification task. It is because, when a negative meme gets identified as a positive meme, it is real problematic situation in our scenario. Therefore, we want to penalize more in this situation and so we would favour precision more than the recall. Putting the value of beta = 0.5 (¡1 to favour precision) in the following formula, we compute the F1 score.

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

The accuracy, precision, recall and F1 scores for our Naive Bayes classifier are shown below in the following table.

Metrics	Scores
Accuracy	88.99%
Precision	0.92
Recall	0.95
F1 Score	93.00%

- **Confusion Matrix** The confusion matrix is determined from our Naive Bayes classifier is shown in Figure 2. It represents the system positive and negative labels in the horizontal axis and actual positive and negative labels in the vertical axis.

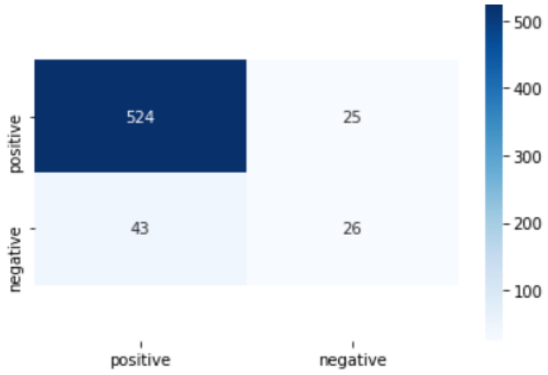

Figure 2: Confusion matrix

- **ROC Curve** From the following ROC curve (Figure 3) of our Naive Bayes classifier, we can see that the True Positive rate is higher and almost towards 1, while the False Positive rate is greater than 0.5, which is mainly due to the imbalance of the class labels in our training data set.

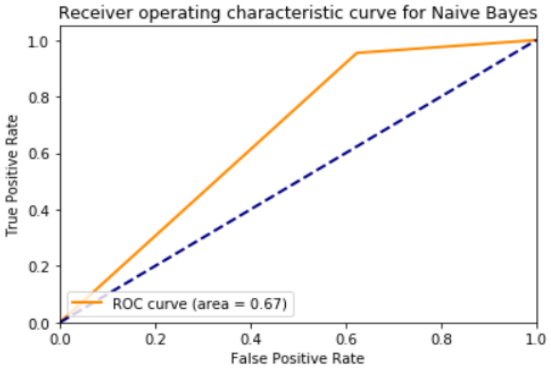

Figure 3: Receiver Operational Characteristics curve

7 CONCLUSION

In this research, we propose a model for the classification of memes into ones having positive and negative sentiments. We take into account the actual image and the text embedded within the image , and feed it into the Naive Bayes classifier for the classification. The accuracy of the system can be improved by treating the imbalance of data between the two classes. This system itself can be improved by training it to recognise the sentiments being expressed, like humour, sarcasm, etc. It can also be trained to identify the degree to which these sentiments are being expressed.

8 ACKNOWLEDGEMENTS

We would like to thank the organisers of SemEval 2020 for providing us with the data needed to train and test the model. We would also like to thank professors Greg Kondark, Denilson Barbosa and Carrie Demmans for their feedback and guidance on making our project better. The report and the project itself was done together and by consulting each other for suggestions on improving the system.

References

Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In *International Conference on Text, Speech and Dialogue*. Springer, pages 196–205.

Vasavi Gajarla and Aditi Gupta. 2015. Emotion detection and sentiment analysis of images. *Georgia Institute of Technology* .