SIS@IIITH at SemEval-2020 Task 8: An Overview of Simple Text Classification Methods for Meme Analysis

Sravani Boinepelli, Manish Shrivastava, Vasudeva Varma
IIIT Hyderabad, India
sravani.boinepelli@research.iiit.ac.in
{m.shrivastava, vv}@iiit.ac.in

Abstract

Memes are steadily taking over the feeds of the public on social media. There is always the threat of malicious users on the internet posting offensive content, even through memes. Hence, the automatic detection of offensive images/memes is imperative along with detection of offensive text. However, this is a much more complex task as it involves both visual cues as well as language understanding and cultural/context knowledge. This paper describes our approach to the task of SemEval-2020 Task 8: Memotion Analysis. We chose to participate only in Task A which dealt with Sentiment Classification, which we formulated as a text classification problem. Through our experiments, we explored multiple training models to evaluate the performance of simple text classification algorithms on the raw text obtained after running OCR on meme images. Our submitted model achieved an accuracy of 72.69% and exceeded the existing baseline's Macro F1 score by 8% on the official test dataset. Apart from describing our official submission, we shall elucidate how different classification models respond to this task.

1 Introduction

Sentiment analysis of online social media has become an increasingly hot topic for researchers and has a wide range of applications from e-commerce, analyzing movie reviews, analysis of current trends, campaigns, etc. Understanding the sentiment bound to a phrase or sentence helps us discern the opinion of a person and a majority of this information is found on online social media such as Instagram, Facebook and Twitter. In concurrence with this, memes shared online are becoming an increasingly popular method for people to express themselves and their opinions, often in the form of a seemingly humorous visual image, on social media. According to Merriam-Webster Online (2020), a meme is an idea, behavior, or style that spreads by means of imitation from person to person within a culture and often carries symbolic meaning representing a particular phenomenon or theme. They are often shared on social media in the form of an image or even a video. This phenomenon poses a great challenge to conventional NLP systems, which are generally purely text based. A hybrid approach is needed to tackle the problem of analysing the sentiment of a meme. It is therefore becoming exceedingly apparent for both the fields of NLP and Computer Vision to rise up to the challenge of adopting multi modal approaches as memes, videos, and other multi modal content are becoming increasingly prevalent online. Through our submission, we aimed to perceive how basic machine learning models performed in this problem space. Our text classification model achieved a Macro-F1 score of 29.65% on the task leaderboard, which exceeded the existing baseline by 8% and ranked 29th among the 566 teams registered which is significant given the simplicity of the model.

2 Related Work

Memes are steadily influencing social media and taking over the feeds of the public. Gal (2015) brought attention to the fact that people use memes to express themselves, and in making them, decide to proclaim

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1190–1194
Barcelona, Spain (Online), December 12, 2020.

their stance on a certain social issue, be it in support or in rejection of the issue. Foster (2014) explored how memes were used in the political context to influence Facebook users during the 2012 Presidential Elections.

There is always the threat of malicious users on the internet posting offensive content, even through memes. A user may come across such distasteful memes and may flag it down, however, it is impossible for them to catch all offensive material at the scale at which multimodal social media grows everyday. Hence, the automatic detection of offensive images/memes is imperative along with detection of offensive text. However, this is a much more complex task as it involves both visual cues as well as language understanding and cultural/context knowledge. Work has been done in the automatic generation of memes (Peirson and Tolunay, 2018) but not much has been done with respect to their analysis and classification of finer aspects like the type and extent of humor and offensiveness incorporated. Ratkiewicz et al. (2010) aims to detect astroturfing in memes on microblog streams via supervised learning based on features extracted from the topology of the diffusion networks, sentiment analysis, and crowdsourced annotations. Ferrara et al. (2013) proposes a fully automatic, unsupervised, and scalable for real-time detection of memes in streaming data by exploring the idea of pre-clustering. Yoon (2016) attempts to analyse Internet memes associated with racism and implements critical discourse analysis in combination with multimodal discourse analysis but there is still a lot of work that needs to be done in the automatic and computational categorization and detection of abusive content in memes.

3 Shared Task Description

The Memotion Analysis Task was divided into three sub-tasks: Sentiment Classification, Humor Classification, and the Scales of Semantic Classes.

The first task (Task A) was to classify a given internet meme as a positive, negative or neutral meme.

Task B was to identify the type of humor expressed by an internet meme. In this task, a meme can be labelled as having more than one category. To work with this, one may have to deal with the consequences of treating the multi labelled data, represented as whole, singular labels of various other labels. Questions of how to categorize memes found in the test data with different sets of labels from those found in the training data would arise.

The third task (Task C) is to quantify the extent to which a particular effect is being expressed. The semantic classes to quantify this extent were given as sarcastic, humorous and offensive and were further to be labelled as "not(0)", "slightly(1)", "mildly(2)", and "very(3)" depending on the extent of the expression.

More details can be found in the task description paper (Sharma et al., 2020).

4 Data

The Memotion analysis task released 8K annotated memes (of which 7K were released for training and the remaining 1K was used for testing) with human-annotated tags, namely sentiment, and type of humor (i.e. sarcastic, humorous, or offensive). The received dataset was a compilation of links to the images and the text results for OCR done on the image. The training data was additionally labelled for how humorous/sarcastic/offensive/motivational a given image was, along with its overall expressed sentiment. The composition of the labelled data was as follows.

Humorous	Sarcastic	Offensive	Motivational	Sentiment
Funny 35%	General 50%	Not 39%	Not 65%	Positive 45%
Very funny 32%	twisted 22%	slight 37%	Motivational 35%	Neutral 31%
Other 33%	Other 28%	Other 24%	–	Other 24%

Table 1: Composition of Memotion analysis training dataset

For Task A, we have formulated the problem as a text classification problem, splitting the given training data into train/dev data as 80% / 20% respectively. The data given was originally categorized as

"Negative", "Very Negative", "Positive", "Very Positive", and "Neutral". This was to be narrowed down further and classified into 3 classes: "Positive", "Neutral" and "Negative".

5 Methodology

In this section, we discuss the ten variations of training models, and their corresponding feature sets, that were used to evaluate the performance of simple text classification algorithms in the given problem space.

5.1 Feature Engineering

In this step, raw text (outputted after running OCR on the meme images) is transformed into feature vectors. We toyed with the following as features.

Count Vectors: The data is represented in the form of a matrix in which every row represents a document from the corpus, every column represents the terms, and every cell represents the frequency count of a particular term in a particular document.

TF-IDF Vectors: TF-IDF vectors have proven to be an effective mechanism to encode textual information into real-valued vectors. It represents the relative importance of a term in the document and the entire corpus. The TF-IDF package from the sklearn library uses count vectorizers to convert text input into a collection of tokens. It allows the flexibility of including higher n-grams in the vocabulary. This can prove to be helpful in classification. We experimented with TF-IDF vectors for different levels of input tokens (words, characters, n-grams).

5.2 Models

The next step in the text classification framework is to train a classifier using the features created in the previous step. We use the Scikit-learn (Pedregosa et al., 2011) and Keras (Chollet and others, 2015) libraries to develop these models. This section offers a brief overview of the various models we employed.

Support Vector Machines: Support Vector Machines (SVM) have long been popular and effective models for multi-class classification problems. We used the sklearn linear SVM library for implementing our SVM based models, taking TF-IDF vectors for inputs of bigrams and tigrams.

Logistic regression: Logistic regression (LR) is one of the simplest ML algorithms that can be used for various classification problems. It is in essence, a statistical model that estimating probabilities using a logistic/sigmoid function. We decided to use this(logistic regression with TF-IDF and character n-grams as features) for our baseline. Hyperparameter tuning proved the model to work best for ranges of unigrams and bigrams.

Random Forest Model: Random forest(RF)s or random decision forests are an ensemble learning method for classification. Our training algorithm applies conventional bootstrap aggregation techniques, or bagging, to tree learners. Random decision forests help correct decision trees' general habit to overfit to the training set. We test their performance against TF-IDF word-level vector and count vector features.

Xtereme Gradient Boosting Model: A boosting model is another version of ensemble model which converts weak learners (like regression, shallow decision trees, etc) to strong ones. The boosting algorithm is popularly used to optimize the performance of decision trees by reducing bias and variance in supervised learning. We test the performance of this model against other baselines with count vectors, word-level TF-IDF and character-level TF-IDF features tuned to ranges of bigrams and trigrams.

CNN: Convolutional neural networks (CNN) are arguably the most popular deep learning architecture. It is computationally efficient, uses special convolution and pooling operations and performs parameter sharing. It is trained on a batch size of 30 with 128 convolution filters and kernel size 5. The convolutional and dense layers use ReLU and softmax activation functions respectively. We use the Adam algorithm (Kingma and Ba, 2014) for network optimization across all our implementations as it is often viewed as a combination of RMSprop and SGD with momentum and can therefore be used for a broad range of tasks like the sentiment classification of meme language.

Bidirectional-LSTM: The Bidirectional-LSTM architecture allows the networks to have both backward and forward information about the sequence at every time step. It is especially helpful for extracting

the context of the input while retaining memory. Its dense layers use ReLU and softmax activation functions respectively and a dropout rate of 0.25 was adopted for our implementation.

CNN-LSTM: The CNN-LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction. It is often used in both the fields of Computer Vision and NLP alike as it shows significant results in both image processing as well as text processing. This architecture is known to be used for the task of generating textual descriptions of images. Much like our CNN model, it is trained on a batch size of 30 with 64 convolution filters, kernel size of 5 and pool length 4. The convolutional and dense layers use ReLU and softmax activation functions respectively. We used this model for our final submission for the task.

6 Results and Analysis

From the results given in Table 2, we can clearly see that the CNN-LSTM architecture presented the best accuracy. This is not surprising given its architecture which is known to be primarily used to solve problems that have a spatial structure to their input which include both the 2D structure or pixels in an image, or the 1D structure of words in a sentence, paragraph, or document. It is therefore, a great starting point to work on processing memes and other images/videos of this sort.

It makes sense that the results received when comparing the accuracy of word-level and character-level features are similar as the data is mostly concise monolingual text(the given memes predominantly used English). Character-level features generally tend to show promising improvement when the data is bilingual/code-mixed. Systems like sub-word level compositions with LSTMs (Prabhu et al., 2016) to capture sentiment at morpheme level are effective to a certain degree, but are limited by their dependence on abundant data that is beyond which was provided for the task. Given that the amount of data is scarce, such deep learning models or use of transformers would fall short as they require an abundance of data to train accurately.

Hyperparameter tuning worked best for unigram or at most bigram parameters. This may have to do with the length of the input text. Models that perform better when the given text input is short in length would perform better when analysing memes due to the inherent structure of a meme. The structure of a meme is such that creators shorten the amount of textual data provided to the extent possible as much of the context for the meme is generally derived from visual input from the image and cultural/general knowledge.

Model	Accuracy	F1 Score
SVM, N-Gram Vectors	41.67%	14.28%
RF, Count Vectors	41.1%	14.93%
RF, Word-Level, TF-IDF	40.31%	15.82%
Xgb, Count Vectors	42.6%	13.3%
Xgb, Word-Level, TF-IDF	42.74%	13.9%
Xgb, Char-Level, TF-IDF	42.74%	14.35%
LR, Char-Level, TF-IDF	41.74%	17.42%
CNN	69.97%	50.87%
Bidirectional-LSTM	71%	52.2%
CNN-LSTM	72.69%	59.04%

Table 2: Performance of various training models for the Memotion Analysis task

7 Conclusion and Future Work

In this paper, we explored and elaborated on various training models for simple text classification algorithms. It is a great starting point to evaluate the performance of basic machine learning models in this problem space. The idea of analysing the sentiment of a meme is a newly blossoming research field with limited work done in the past. It therefore holds a lot of promise.

As much of the context for the meme is generally derived from visual input from the image and general knowledge, future work may consider the possibility of looking into a hybrid model that could process the image as well as the text to give more context to the OCR outputted data. It is crucial to keep in mind limitations such as the limited text associated with memes. We may also look into methods to procure the cultural/general knowledge needed to understand the meme as well.

References

François Chollet et al. 2015. Keras. https://keras.io.

E. Ferrara, M. JafariAsbagh, O. Varol, V. Qazvinian, F. Menczer, and A. Flammini. 2013. Clustering memes in social media. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pages 548–555.

Bobbie J. Foster. 2014. It's all in a meme: A content analysis of memes posted to 2012 presidential election facebook pages.

Noam Gal. 2015. "it gets better": Internet memes and the construction of collective identity. *New Media amp Society*, 01.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Merriam-Webster Online. 2020. Merriam-Webster Online Dictionary. http://www.merriam-webster.com.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Abel L. Peirson and E. Meltem Tolunay. 2018. Dank learning: Generating memes using deep neural networks. *CoRR*, abs/1806.04510.

Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. 11.

Jacob Ratkiewicz, Michael D. Conover, Mark R. Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. 2010. Detecting and tracking the spread of astroturf memes in microblog streams. *CoRR*, abs/1011.3768.

Chhavi Sharma, Deepesh Bhageria, William Paka, Scott, Srinivas P Y K L, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 Task 8: Memotion Analysis-The Visuo-Lingual Metaphor! In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain, Sep. Association for Computational Linguistics.

Injeong Yoon. 2016. Why is it not just a joke? analysis of internet memes associated with racism and hidden ideology of colorblindness.