

Identification of Persuasive Techniques Within Memes

Aditya Thakur	Taylor Short
Adt.10@unb.ca	tshort1@unb.ca
3740903	3586404
Patrick Mockler	
pmockler@unb.ca	
3434533	

ABSTRACT

As a unique relic of internet pop culture, memes have been used for decades to communicate, share humor, and connect humans over a virtual world. They have also been increasingly subjected to polarization and political narratives. Recent years have proven that social media platforms perpetuate disinformation, spread propaganda, and generate divisive narratives. It is important to identify and relay persuasive language to viewers who can then make informed decisions on the content they interact with. By analysing the textual content of internet memes, we can identify and classify memes under 20 persuasive language techniques. We use a multi output classifier with LinearSVC to classify memes based on their persuasive language techniques. We compare LinearSVC to a baseline multilabel Logistic Regression algorithm and compare the micro f1-scores of each. Both models perform poorly on our prediction task with around 0.2 total accuracy and around 0.35 micro averaged f1-scores. We discuss some challenges hindering our approach and identify opportunities for further improving performance.

KEYWORDS

Natural Language Processing; Propaganda; Meme; Social Media; Persuasive Language

1. INTRODUCTION

In the age of the internet, social media has become a popular location for sharing and discussing news, opinions, and communities. As a virtual town square, social media allows its users to gather with likeminded individuals or clash with those who are against them. This can lead to polarized politics and toxic behavior within online spaces. People use these platforms to spread propaganda and further their ideologies. Memes are a prevalent way of communicating over social media, using popular media and comedy to relate with others and convey ideas. While some memes may be strictly comedic or nonsensical, many of them attempt to spread a message using persuasive language.

Our project focuses on classifying memes based on their textual content, identifying which of the 20 different techniques of persuasive language apply. Examples include name-calling, smears, loaded language, and straw-mans. This is based off the SemEval 2024 Task 4 [1], using their curated dataset. Social media can sway public opinion on a massive scale. Being able to identify the use of persuasive language within memes can help track and understand the proliferation of propaganda and misinformation online.

2. BACKGROUND

2.1 Memes

Memes are pieces of media (images, videos, texts, music, etc.) that generally serve a comedic, relatable, or informative purpose. These memes are spread over social media, some becoming so popular in use that they add to a common vernacular shared between internet users.

2.2 Persuasive Techniques

22 different persuasive techniques are defined for the SemEval task with 20 specifically applying to the text-based subtask. These contain logical fallacies and rhetoric designed to push an agenda within a meme. The classes are as follows: flag-waving, glittering generalities, loaded language, straw man, name calling, obfuscation, red herring, reductio ad hitlerum, repetition, slogans, smears, thought-terminating cliché, and whataboutism. These are the labels that our model must accurately predict [1].

2.3 Multi-label Classification

The memes used in this task typically do not have only one persuasive technique applicable. The same meme can use multiple techniques. This means that the task requires multi-label classification where the labels are not mutually exclusive. This makes it different from multi-class classification in which samples can only belong in one class. This is an important distinction because it requires different methods of learning and evaluation. [2]

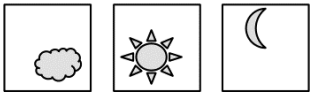

	Multi-Class	Multi-Label
C = 3	Samples  Labels (t) [0 0 1] [1 0 0] [0 1 0]	Samples  Labels (t) [1 0 1] [0 1 0] [1 1 1]

Figure 1. A comparison between multi-class and multi-label classification. [2]

3. METHODOLOGY

To properly predict the persuasive language techniques based on textual meme content, we need the ability to predict multiple output classes. A sample can contain multiple persuasive techniques or contain no persuasive techniques. The dataset contains a set of text content taken from meme images along with the accompanying labels. Internet memes are typically structured as a photograph with an overlay of text that combine to convey a message. Removing the image portion of the meme and rendering only textual content leaves short segments of text often lacking in context.

3.1. Tokenization

These short-form text segments can be a challenge to process, so our first approach to predicting labels from these contextless segments is to carefully tokenize the text with the understanding of the underlying structure of a meme. Memes can have single-line text overlayed at the top of an image or can be multi-line and cover both the top and bottom of the image. The format of the text within the meme could contribute to the persuasiveness of the meme

and may indicate a certain technique. For this reason, it is important to maintain newlines and punctuation during the tokenization process.

3.2. Models and Baselines

As a multilabel classification problem, we need to ensure that every potential target is assessed to determine if it is probable for the text content. For this classification problem, we wanted to compare a regression-based model against a classification model. Logistic Regression was chosen as the baseline since it is a well-known algorithm and able to predict multiple output classes for each target to accommodate our persuasive language detection task. To choose our comparison model, we looked at similar studies comparing traditional algorithms to identify trends in performance. We found that Support Vector Machine performed better than Logistic Regression on both a binary and a multilabel classification task [3, 4]. We initially tested LinearSVC and SVC on our dataset and found that LinearSVC performed slightly better, so we opted for LinearSVC as our comparison to Logistic Regression. Both Logistic Regression and LinearSVC use a One Vs Rest approach to predicting targets [5]. Each target label is treated as a binary classification and the probability of that target is predicted, enabling multiple targets to be predicted for a single data sample [5].

3.3. Benchmarks

Multilabel data has a unique consideration when considering benchmarks and evaluation approaches. For a given sample to be a correct prediction, all predicted labels must match the gold standard [5]. Consequently, an overall accuracy would be expected to be lower than individual class accuracies that look only at a single target prediction. We can use a class's individual F1 score to identify trends in class accuracy. Section 4.2 shows the variation in our sample dataset and shows that there is no clear sample bias towards a single class. Precision and recall can also be helpful for understanding the performance of each individual class's classifier. Since we deal with multilabel data, we can use a micro-F1 score to obtain an overall view of the model's performance that holds equal weight for each prediction.

4. IMPLEMENTATION DETAILS

4.1 Selection of Libraries

For implementation of our project, we made use of the SciKit-Learn(sklearn) library, which is an open-source and powerful machine-learning toolkit used for training and building models [5]. It is extensively used in the industry and provides rich set of tools for data processing, feature engineering and contains a wide array of machine learning algorithms, making it our optimal choice of selection. For processing our data, we have incorporated the Natural language Toolkit (NLTK) [5], a powerful library for natural language processing (NLP) tasks. It is well-known for handling various aspects of NLP such as tokenization, stemming to part-of-speech tagging and sentiment analysis. For our purposes, we used this library for tokenization of our data. We also used Pandas [8] and Matplotlib [9] libraries for doing data analysis. By converting the training dataset (Json file) into Dataframe, which is an in-built data structure, we were able to manipulate and clean the data.

4.2 Dataset

The dataset [1] we chose for our NLP research project contains contextual information on memes. It consists of id, text, labels, and links. For our purpose, we are only concerned with the text and the labels as they represent the input and target variables of our dataset. The text contains the textual content of the memes, and its corresponding labels

are a list of persuasion techniques. Figure 2 represents the count value of the labels. There are 7000 rows of values in the dataset out of which 1264 rows have labels which have an empty list.

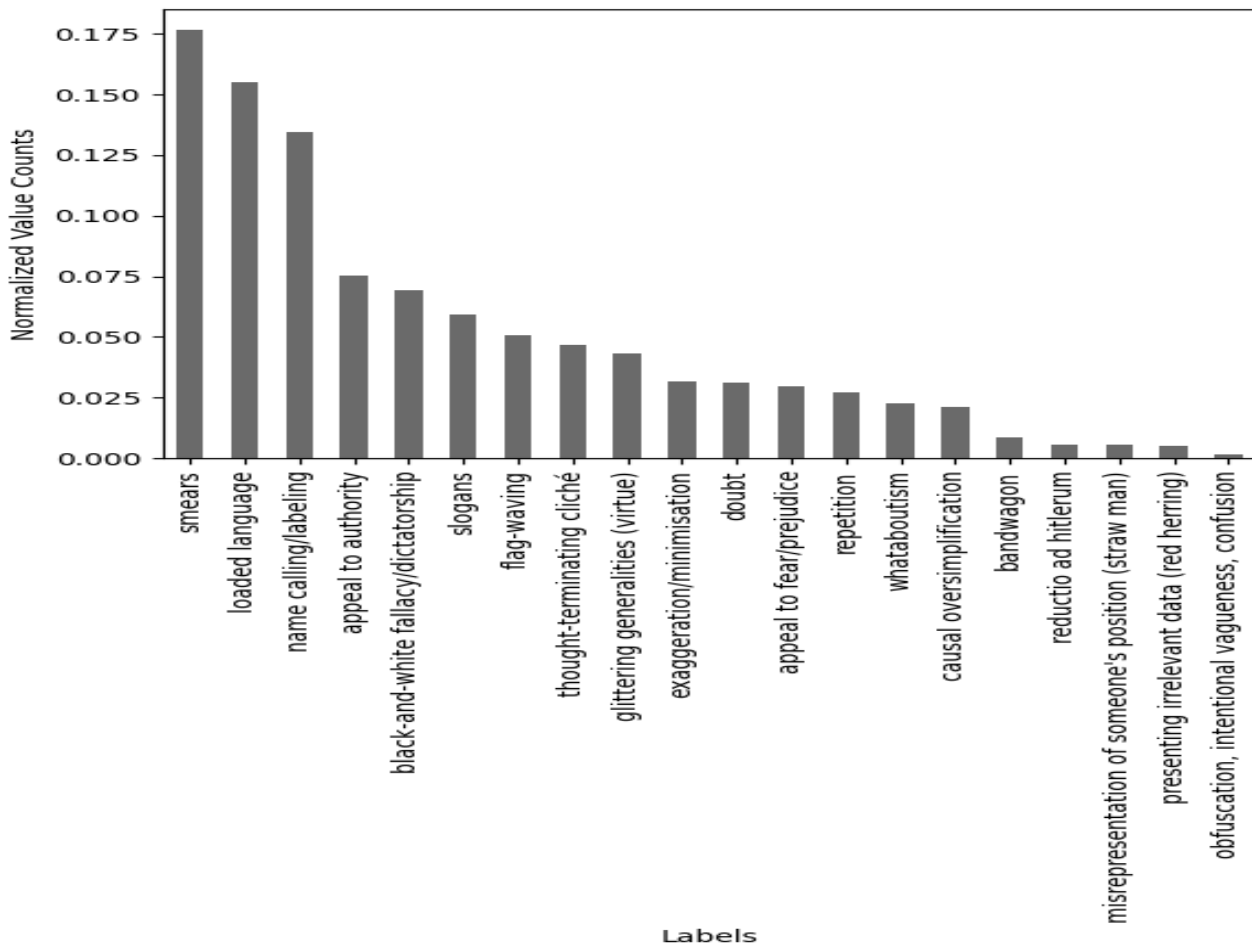


Figure 2. A graph displaying the frequency of each label within the dataset.

4.3 Workflow and Design

4.3.1 Tokenization and Data Analysis

In the process of preparing our dataset for analysis, we implemented tokenization on both the text and the labels fields. In our tokenization, we applied lowercase to both fields and split the text wherever “\n” occurred. By using NLTK library [6] and re(Regular Expression) [7], we were able to remove the stop words and lemmatize the text which would help in getting more accurate results. The resulting tokenized data was then stored in the “train_tokenized” JSON file and the rest of the analysis was carried out using this file. In this way, we ensure to keep the original raw data separated from further implementations. We further analysed the data by plotting a graph, cleaning the data, and finding missing values.

4.3.2 Implementing Multi-Classification Models

We implemented multi-output logistic regression as our primary baseline by importing scikit-learn’s MultiOutputRegressor as the wrapper class [5]. Logistic regression is well-suited for multiclass classification tasks, making it an appropriate choice for choosing as our baseline. This classifier serves as the foundation of our

subsequent experiments. To get the results from the classifier we implemented `accuracy_score` and `classification_report` from `scikit-learn` [5]. After successfully implementing the Logistic Regression baseline, we implemented Linear Support Vector Classification (SVC) from the `scikit-learn` [5]. SVC is well-known for handling complex classification problems and `MultiOutputClassifier` can be applied to independently handle multiple outputs [5]. Alternatively, we implemented One Vs Rest classifier using these same models.

5. EVALUATION

Table 1. A comparison of the performance of Logistic Regression and LinearSVC based on f1-score.

label	f1-score (LR)	f1-score (LinearSVC)
appeal to authority	0.40	0.44
appeal to fear/prejudice	0.11	0.10
bandwagon	0.11	0.33
black-and-white fallacy/dictatorship	0.20	0.22
causal oversimplification	0.07	0.18
doubt	0.16	0.12
exaggeration/minimisation	0.16	0.17
flag-waving	0.38	0.40
glittering generalities (virtue)	0.20	0.22
loaded language	0.46	0.50
misrepresentation of someone's position (straw man)	0.00	0.00
name calling/labeling	0.38	0.42
obfuscation, intentional vagueness, confusion	0.00	0.00
presenting irrelevant data (red herring)	0.00	0.00
reductio ad hitlerum	0.00	0.00
repetition	0.17	0.21
slogans	0.28	0.30
smears	0.47	0.47
thought-terminating cliché	0.16	0.18
whataboutism	0.10	0.14
Micro-Averaged	0.34	0.36

5.1. Experimental Setup

The experiment uses the dataset of textual meme content and their accompanying persuasive language technique labels in the form of a JSON file. The textual content is processed so the text is rendered in all lowercase, newlines are tokenized, and stop words are removed. We also lemmatize the tokens. The data set, originally of size 7000, is split into ~4000 training samples and ~3000 test samples. Labels are binarized for the one vs rest prediction models. The logistic regression model is trained with a max iteration size of 1000 and evaluated with a multi output regressor to predict multiple output classes. The LinearSVC model is trained with a max iteration size of 5000 and uses a

multi output classifier method. For each model, an evaluation report is generated comparing the precision, recall, and f1-scores of each predicted class. The models are finally scored on the micro-averaged f1-score.

5.2. Evaluation Metrics Section

For the One Vs Rest prediction models, we look at precision, recall, and f1-scores of each individual class to provide insight on the performance of the classifiers. For our final evaluation, compare use the f1-scores of each class for our two models. For overall performance, we compare the micro-averaged f1-scores. The overall accuracy of each model is not a suitable metric for understanding model performance because a sample prediction is only deemed correct if all predicted labels on that sample are accurate. Our total accuracy would be low compared to class accuracies due to cases where most labels are correctly predicted but one label is incorrect. A micro-averaged f1-score enables us to evaluate the overall accuracy through each individual label prediction.

5.3. Experimental Results

Both Logistic Regression and LinearSVC performed similarly. Some labels were predicted more accurately such as appeal to authority, smears, and loaded language. These were all within an f1-score of 0.40 to 0.50. Other labels with less representation in the dataset saw much lower f1-scores such as red herrings, reductio ad hitlerum, and straw mans. These all achieved an f1-score of 0. Overall LinearSVC performed very slightly better with a micro-averaged f1-score of 0.36 while Logistic Regression achieved a micro-average of 0.34. Both models performed poorly and require some modification to effectively label these memes and overcome the challenges that the dataset and task provided.

6. CHALLENGES

Our goal was to evaluate and compare two models for predicting persuasive language in the textual content of memes. A few major challenges resulted in our overall low model performance. We selected subtask 1 of task 4 from SemEval 2024, which involved using only the textual content of memes [1]. Additional subtasks in this task set perform machine learning on the images to further identify which persuasive techniques are being used [1]. A significant amount of context is lost when only considering the textual content of these memes, rendering natural language processing difficult for this task. Additionally, memes are typically short and easy to read. The textual content surrounding memes tends to be very short and lacking in additional information or explanation, often requiring the reader to read between the lines. This poses a considerable challenge for models trying to classify the techniques since the content attempts to portray a complex topic in a short segment of text.

A technical challenge that we faced during the implementation was trying to moderate the low accuracy scores of our models. The nature of multilabel classification problems significantly reduces the overall accuracy of our models despite having a higher accuracy across classes. This is because one or two wrong class predictions in a multilabel sample will render the entire sample prediction incorrect, reducing the overall accuracy. To explore this concept, we tried implementing a few additional models. We implemented SVC, LinearSVC, and Logistic Regression models. We also tried an alternative method by using a One Vs Rest model method from sklearn based on each of these classifiers. We ultimately kept Logistic Regression and LinearSVC for performance. We also experimented with different variations in tokenization, like lemmatization, removing stopwords, and tokenizing punctuation. A final experiment we attempted was altering our dataset by removing empty label sets. In one approach, we replaced empty datasets with a new “normal” label to see if we could accurately predict samples without persuasive language techniques. For another approach, we modified the dataset to remove empty labels entirely and train only on positive

persuasive language samples. Neither of these approaches proved successful in improving accuracy. Despite the low final performance of our models, we were able to explore different approaches to natural language processing. This also gave us an opportunity to deeply consider some of the major challenges around context-based natural language processing.

7. INDIVIDUAL CONTRIBUTION

Taylor Short: Taylor assisted in task selection and setting up the account to retrieve the datasets. Also contributed file I/O, data tokenization, and the One Vs Rest alternative modelling functions to the codebase. Taylor wrote the Abstract, Methodology, Evaluation, and Challenges, and Conclusion sections of the report.

Patrick Mockler: Patrick assisted in task selection, planning, research and writing for the initial proposal as well as managing / editing the final report and contributing the intro, background, evaluation table, experimental results, and ethics section of the report.

Aditya Thakur: Aditya assisted in task selection and contributed in file I/O, data tokenization, data analysis, Multi-Output Logistic Regression, LinearSVC, results, and organizing code structure. Aditya wrote the Implementation details section of the report.

7. CONCLUSION

For the task of classifying persuasive language techniques in meme textual content, both our Linear Regression baseline and our LinearSVC models performed poorly overall. The missing context of the image content in the meme poses challenges to accurately classifying the data. Additionally, we found that most of the data consisted of small text segments which would not be expected to train a model sufficiently. We also identified that the size of the dataset and the uneven distribution of label samples impacted our models' ability to classify certain classes correctly. With an improved dataset consisting of more samples and a more evenly distributed set, we expect to see better results in classifying the data on text alone. Despite the challenges, our baseline multi output Logistic Regression model proved the most successful on overall accuracy, but less successful on micro-averaged f1-scores. Both models' scores were within a few percentage points. Given more time, we would have liked to apply deep learning and compare our two traditional model implementations to a more advanced neural network. Future work could also be performed into data augmentation or further data collection to provide a more robust dataset to train our models on.

9. ETHICS STATEMENT

While the task of identifying propaganda and misinformation in memes can be helpful to research and study the spread of this kind of internet content, the technology used for this purpose has the potential to also be used for automated mass censorship by online platforms. This could cause major social harm by silencing minority voices whose politics and conduct go against the interest of the social media companies. What constitutes as harmful propaganda or misinformation tends to be somewhat subjective and delegating private social media companies as the arbiters of truth could greatly skew what opinions or beliefs are acceptable online.

Furthermore, the potential of bias within the trained model could lead to harsher detection for specific political groups over others. Steps would need to be taken to ensure that the model treats memes of diverse origins fairly and that none are given preferential treatment. The names and likenesses used in the dataset could be cause for concern.

For example, the name Donald Trump occurs frequently within the training data, but the machine learning data should not give weight to the inclusion of an individual's name when discerning the persuasive language being used.

10. REFERENCES

- [1] Dimitrov, D. *et al.* (no date) *Semeval 2024 task 4 'multilingual detection of persuasion techniques in memes'*, *SemEval2024 shared task on 'Multilingual Detection of Persuasion Techniques in Memes'*. Available at: <https://propaganda.math.unipd.it/semeval2024task4/> (Accessed: 30 November 2023).
- [2] Raúl Gómez Bruballa. 2018. '*Understanding categorical cross-entropy loss, binary cross-entropy loss, Softmax loss, logistic loss, focal loss and all those confusing names*'. Available at: https://gombru.github.io/2018/05/23/cross_entropy_loss/ (Accessed 29 November 2023)
- [3] Kamath, C. N., Bukhari, S. S., and Dengel, A. Comparative study between traditional machine learning and deep learning approaches for text classification. In *Proceedings of the ACM Symposium on Document Engineering 2018* (New York, NY, USA, 2018), DocEng '18, Association for Computing Machinery.
- [4] Morales-Hernandez, R. C., Jagnuey, J. G., and Becerra-Alonso, D. A comparison of multi-label text classification models in research articles labeled with sustainable development goals. *IEEE Access* 10 (2022), 123534–123548.
- [5] Scikit-learn: Machine Learning in Python, Pedregosa et al., *JMLR* 12, pp. 2825-2830, 2011.
- [6] NLTK :: *Natural Language Toolkit*. Retrieved November 30, 2023 from <https://www.nltk.org/>
- [7] Van Rossum, G. The Python Library Reference, release 3.8.2. *Python Software Foundation*, 2020.
- [8] Wes McKinney. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* (2010), Stefan van der Walt and Jarrod Millman, Eds., pp. 56 – 61.
- [9] Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95