Multi-Modal Classification for Polarization Intent Detection in Social Media

Tobby Lie
tobby.lie@ucdenver.edu
University of Colorado Denver
Denver, Colorado

Haadi Jafarian
haadi.jafarian@ucdenver.edu
University of Colorado Denver
Denver, Colorado

Stephen Hartnett
stephen.hartnett@ucdenver.edu
University of Colorado Denver
Denver, Colorado

Hamilton Bean
hamilton.bean@ucdenver.edu
University of Colorado Denver
Denver, Colorado

Farnoush Banaei-Kashani
farnoush.banaei-kashani@ucdenver.edu
University of Colorado Denver
Denver, Colorado

Abstract

The ease of access to social media has made it extremely easy for malicious agents to disseminate divisive information on a mass scale. With the widespread reach of messages with various intents it becomes more difficult for experts to analyze such information in a timely and accurate manner to detect these intents. Within the research community, capturing online threats of this nature has been a growing area of interest. In this paper, we develop a methodology to detect intention of social polarization information shared by a single agent in social media. Toward this end, we use and study the Internet Agency Facebook ads dataset released by U.S. House Intelligence Committee. We train and evaluate a series of models for text- and image-based intention detection and show that a multi-modal model that uses both data modalities outperforms other unimodal models.

1 Introduction

Social media has made the spread of information with various intents increasingly accessible to a wide array of users. In this collection of participants, there are inevitably those with ill intentions, e.g., intent to segregate online groups. With zero barriers of entry to participate in this cycle of information sharing, many unintentionally amplify this issue. In this paper, we examine the specific case of alleged Russian disinformation in the form of Internet Research Agency ads to introduce a methodology for automated detection of polarization intent in information shared by a single agent in social media. Thereby, we are able to provide a scalable solution for experts in the field in the form of an open-source tool [1].

Toward this end, we have labeled the Internet Research Agency (IRA) Facebook ads dataset released by U.S. House Intelligence Committee to identify whether each ad is encouraging "Violence" or "Apathy", to polarize the U.S. society in these two directions [2]. Accordingly, we developed a number of machine learning and deep learning models based on text and image data included in IRA ads as tools to automate the detection of polarization intention in data such as the IRA dataset. For example, this tool can be used as a pretrained model for transfer learning to automate detection of polarization intention in datasets such as COVID-19 Twitter dataset [6] or CoronaVirusFact Alliance dataset [1] to identify polarization intents in these so-called "infodemics".

Similar research efforts in this specific area investigate polarization on social media studies, intent classification of short-text on social media, and image and text feature fusion for intention detection on Instagram posts. Our work differs from these prior works as we deliver a methodology that encompasses all of these areas in a comprehensive manner rather than individually studying each topic.

Our contribution lies in the multi-modal approach of classifying Facebook ads based on textual data, image data and a fusion of both. By developing high performing models to this end, our proposed tool is able to detect polarization attempted by a single agent via revealing conflicting intents. We have been able to create highly successful models for single-modal and multi-modal features. We have obtained recall scores ranging from 0.95-1.0 and F1 ranging from 0.91-0.97 with our most successful implementations.

The rest of this paper is organized as follows. In Section 2 we explore work related to our own. We review our dataset in detail in Section 3. In Section 4, we present the methods applied to reach our goals. Section 5 discusses our experiments and results, and in Section 6, we provide a conclusion as well as ideas on future work that builds upon what we have accomplished.

2 Related Work

In this section we review two distinct categories of work related to our focus: polarization detection and intent detection based on multi-modal models.

[1] Lie, Tobby. "Tobby-Lie/Russian-Disinformation-Project." GitHub, 12 Dec. 2019, github.com/tobby-lie/Russian-Disinformation-Project.

Garimella et al. [4] survey polarization and how it is manifested online and in particular on social media. In this work they explore the concept of polarization drawing from social and political sciences. Further, they examine the algorithmic techniques for detection, quantification, and mitigation of polarization. They investigate various ways of identifying and quantifying polarization through content vs. network-based methods, define mitigation methods for bursting the filter bubble, study methods for visualizing the ideology space, and explore the intervention of algorithmic methods.

Valerie et al. [9] approached the problem of intent multi-class classification from the perspective of social data generated in times of crisis events. They utilized a hybrid feature representation done by combining top-down processing using knowledge-guided patterns and bottom-up processing using a bag-of-words model. While we do implement similar tokenization strategies, our methods go further in pre-processing via contraction mappings as well as data sampling in various ways to handle class imbalance. Additionally, our end results were a product of multi-modal classification which is more extensive than just textual data. Sikka et al.[3] present a review of approaches in handling multi-modal datasets such as the dataset we explored in our study. This work begins with the examination of the relationships between textual and image data. Previous work expanded upon in this review that have studied this relationship include visual question answering, extracting literal or connotative meaning from a post, the role of image as context for interaction and pragmatics in dialog and prompts for user-generated descriptions and intention-focused prediction for politician portrayals in news media. Furthermore, they focus on a specific case study to detect the intent of Instagram posts in eight categories: advocative, promotive, exhibitionist, expressive, informative, entertainment, provocative/discrimination and provocative/controversial. They look into contextual taxonomy to measure the relationship between the captions and images of their Multimodal Document Intent Dataset. Our work is different from this work as we focus on developing a generic pretrained model for polarization intent detection.

3 Dataset

We now elaborate on the data we worked with as formerly touched on. Our dataset is released by the U.S. House Intelligence Committee and consists of 3012 data samples in the form of Facebook ads that the Internet Research Agency (IRA), a Russia-linked "troll farm" purchased leading up to the 2016 election campaign. Each data sample consists of 24 pieces of key metadata information. The most notable pieces of information we considered consisted of the URL to the original ad, an identifier, a title, a description, an image URL and tags describing each entry. In particular, we closely examined the "anger" and "fear" tags as a basis for our class labels, namely "Violence" and "Apathy". 160 students from the "Interpreting Strategic Discourse" class offered by the University of Maryland's Department of Communication hand-coded the dataset. [7]

We made use of description texts and image URLs from features and utilized the tag data as class labels. The text data was extracted from the ads themselves and the image data contained the entire ad. The tag data consisted of metadata keywords that described what each ad contained.

4 Intent Detection

In this section we examine the methods we employed which ranges from data preprocessing to model generation. Our approach was to first develop intent classifiers for text and image data separately. Next, we fused the two modalities into one feature space for integrated classification.

4.1 Text Classification

For text classification we studied both classic machine learning models as well as more recent text embedding based solutions with deep learning models. For classic machine learning models, we utilized the description field of each sample which consisted of the words extracted from the ads themselves. For each sample, we utilized a bag of words method to convert the text to a machine-readable format conducive to machine learning models. We ran scikit-learn's CountVectorizer in order to tokenize our text documents. We further implemented a contraction mapping with Python library spaCy in order to further clean up our textual data. Undersampling and oversampling were implemented on our datasets for each of the two classifications described previously, as the negative instances of our labels far outweighed the positive instances ("anger" and "fear"). For undersampling, we utilized random data frame manipulations via pandas to reduce the number of majority class instances to that of the number of minority class instances. For oversampling, we utilized RandomOverSampler from Python library imblearn.

After preprocessing our description text data, we trained two classic machine learning classifiers: support vector machine with linear kernel and Naive Bayes via scikit-learn on the two classification tasks for the original imbalanced data, undersampled data and oversampled data on a train test split of .80 train and .20 test. After fitting our support vector machine and Naive Bayes to our training data, we ran predictions on our testing data, and for each case mentioned above, we derived accuracy, precision, recall and F1 scores as well as a confusion matrix.

Google's BERT (Bidirectional Encoder Representations from Transformers) was used for implementing a sentence embedding as a means of feature extraction from our textual description data. With the aid of `torch`, a pretrained BERT model and a BERT tokenizer, we were able to achieve this. The pretrained base model released by Google ran for four days on four cloud TPUs on Wikipedia and a corpus of over 10,000 books of varying genres. We utilized a basic BERT model with no specific output task which has proved to be useful for embeddings. A special BERT tokenizer was utilized in order to convert our text data into a format that the model accepted. In order to extract embeddings from our text, we used torch to manipulate our data as tensors to feed into our pretrained model. BERT returns an output of four dimensions: a layer number with 13 layers of the architecture in total, a batch number that represents the text fed in, the word/token number that varies based on the input and finally, the hidden unit that consists of 768 features. We then used the generated embeddings for text classification.

4.2 Image Classification

For our image data, we extracted each image from the provided image URLs and stored them locally. For each training period, we loaded our images from local storage in a random order by shuffling our image paths. We then utilized OpenCV and Keras in order to resize our images to acceptable dimensions and convert them to arrays which is required for our neural networks. In order to deal with the class imbalance in this context we simply reduced and duplicated our data locally for undersampling and oversampling respectively and utilized ImageDataGenerator from Keras in order to maintain variance in our data even in the case of duplicated instances to prevent overfitting.

We trained multiple neural networks via Keras for various training periods to get a sense of which models performed best on our image dataset; these results will be displayed in detail in Section 5. We trained our models via a tensorflow-gpu backend to take advantage of GPU compute power. By utilizing the fit generator in Keras, our ImageDataGenerator could create variance in each training sample on the fly. We implemented two primary models in the training process: ResNet50 [8] and a scaled-down VGG architecture denoted as SmallerVGGNet [10]. Our initial tests revolved around a single classification task, that being “Violence” in order to get a baseline for how our models performed before extending this work to “Apathy”. After each training period, a similar process of deriving classification metrics and a confusion matrix was completed in order to evaluate the success of our models. For our initial stage of “Violence” classification testing, we varied the training periods, utilized class weighing, undersampling and oversampling.

4.3 Classification via Fusion of Modalities

To make the most use of our text embedding output, we desired a single vector output for easy fusion with our image embedding later on. To do this, we averaged the second to last hidden layer of each token thus, producing a single 768 length vector; each of these outputs was saved to a list of numpy arrays in .npz format for later use.

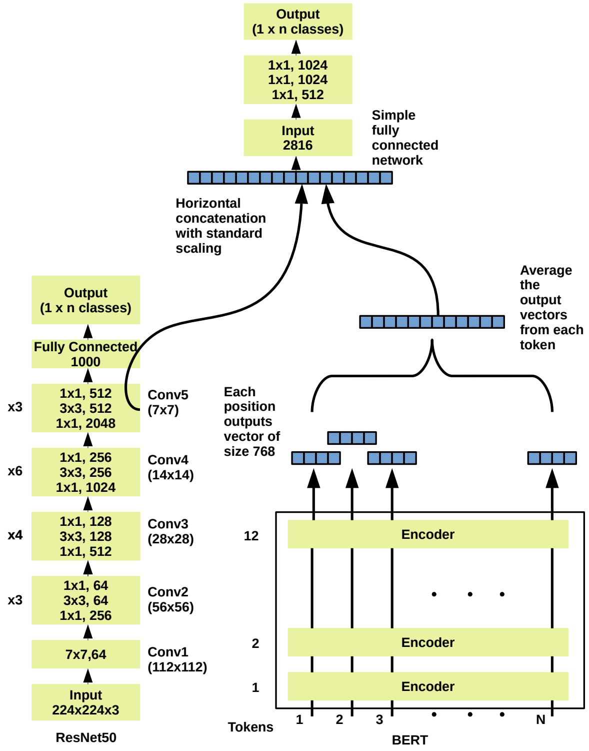

Figure 1. High-Level Representation of Fusion Architecture

To perform image embedding, for each case of our classification tasks, we utilized the designated ResNet50 model that we trained for that specific task. We employed the same preprocessing methods as mentioned previously for image classification to prepare our images for embedding. Once this was complete, we ran each image through our pre-trained ResNet50 and extracted the features in the last average pooling layer of the model. This yielded a single 2048 length vector for each image input which we saved as a list of numpy arrays in .npz format for future fusion.

Beyond single modality training on text and image data individually using separate classifiers, we also utilized the previously mentioned word and image embedding techniques to implement a fusion of our different modes of data in an effort to improve classification performance.

We implemented a simple fusion strategy that allowed for two feature vectors of different lengths to fit together. Our approach was a horizontal concatenation of the two embedding vectors derived from the text and image embedding techniques described above, followed by the use of StandardScaler from scikit-learn in order to normalize our newly-integrated vector. After this fusion, we handled the issue of imbalanced data by simple numpy array operations to randomly oversample our dataset as this sampling method proved to be the most successful for all previous experiments when compared against original imbalanced data and undersampling methods.

We developed a simple, fully-connected neural network to feed in our newly-sized vectors for classification. Our architecture consisted of four dense layers consisting of 1024 neurons using relu activation in the first two layers, 512 neurons in the third layer using relu activation and a final softmax layer. We trained this model with the newly-fused features on an .80 to .20 validation split and derived classification metrics and a confusion matrix for each test run.

Figure 1 contains a high-level diagram of the fusion architecture explained in this section.

5 Experimental Analysis

In this section we dive deeper into the specific experiments we ran on our data via our models and present our quantitative results.

Our experiments were set up in separate Jupyter Notebook environments and harnessed the computing power of a Geforce RTX 2070 Super for training neural networks. We chose Python 3 as our language and Conda was leveraged in order to handle dependencies via virtual environments.

For each experiment, there was an issue of class imbalance present which we dealt with by utilizing oversampling and undersampling strategies with the addition of adjusting class weights for our neural network experiments. We derived accuracy (Acc.), precision, (Prec.), recall (Rec.) and F1 scores across all tests to compare the performances of our models.

We utilized a support vector machine and Naive Bayes for our text classification and derived our metrics through predictions on a test split. Observing the results from Table 1, we can clearly see that for both classification tasks, using a support vector machine on oversampled data handled the issue of data imbalance very well across the board as denoted by the “Oversampled Violence” entries for the “Support Vector Machine” results. From the Naive Bayes results we can observe a similar trend in that oversampling handled the data better in almost every measure except for an equal accuracy between “Imbalanced” and “Oversampled Violence” and the “Imbalanced” data yielding better results in accuracy for “Apathy”. For the most part oversampling was capable of yielding the best performing metrics. Our experiments for textual data classification required that we attempted different sampling strategies to come to the conclusion that oversampling proved to be the most effective way for us to deal with class imbalance for our data specifically, and so we extended this thinking to our image data and fused feature data. We can also conclude from the two text classifiers that the support vector machine trained on oversampled data outperformed the Naive Bayes model in nearly every aspect and consistently did better on its oversampled data than in any of its own other test cases.

Table 1. Performance Results from our Varying Modality Experiments

Support Vector Machine	Acc.	Prec.	Rec.	F1
Imbalanced Violence	0.74	0.45	0.51	0.48
Undersampled Violence	0.69	0.64	0.72	0.68
Oversampled Violence	0.80	0.85	0.79	0.82
Imbalanced Apathy	0.88	0.32	0.38	0.35
Undersampled Apathy	0.66	0.65	0.63	0.64
Oversampled Apathy	0.91	0.95	0.87	0.91
Naive Bayes				
Imbalanced Violence	0.77	0.54	0.58	0.56
Undersampled Violence	0.73	0.82	0.70	0.76
Oversampled Violence	0.77	**0.89**	0.73	0.80
Imbalanced Apathy	0.90	0.23	0.55	0.32
Undersampled Apathy	0.73	0.76	0.69	0.72
Oversampled Apathy	0.85	0.96	0.79	0.87
SmallerVGGNet				
Imbalanced Violence	0.67	0.78	0.75	0.77
Adjusted Class Weights Violence	0.56	0.94	0.0.43	0.59
Undersampled Violence	0.56	0.71	0.27	0.39
Oversampled Violence	0.75	0.73	0.77	0.75
Imbalanced Apathy	0.88	0.88	**1.0**	0.94
Adjusted Class Weights Apathy	0.88	0.88	**1.0**	0.94
Undersampled Apathy	0.58	0.58	0.65	0.61
Oversampled Apathy	0.64	0.63	0.72	0.67
ResNet50				
Imbalanced Violence	0.76	0.79	0.90	0.84
Undersampled Violence	0.60	0.77	0.35	0.48
Oversampled Violence	0.87	0.87	0.87	0.87
Imbalanced Apathy	0.93	0.99	0.93	0.96
Undersampled Apathy	0.92	0.95	0.89	0.92
Oversampled Apathy	0.95	**0.99**	0.92	0.95
Multi-Modal Fusion				
Imbalanced Violence	0.86	0.76	0.70	0.73
Undersampled Violence	0.85	0.82	0.89	0.85
Oversampled Violence	**0.91**	0.88	**0.95**	**0.91**
Imbalanced Apathy	0.92	0.57	0.63	0.60
Undersampled Apathy	0.82	0.77	0.89	0.82
Oversampled Apathy	**0.97**	0.94	**1.0**	**0.97**

To gain an understanding of which neural network architecture might work best for image classification, we ran a series of initial experiments on “Violence” classification via a combination of varying architectures, training periods and sampling methods. We first began by using SmallerVGGNet to benchmark the various sampling methods coupled with different lengths of training time in order to understand which combination of these two variables provided the most optimal results. Notice also that for neural networks we can adjust the weights of each class for training rather than having to undersample or oversample; we later found that this was not useful for SmallerVGGNet and did not extend this to ResNet50. When making the jump from SmallerVGGNet to ResNet50 we decided to test ResNet50 without class weight adjustment as this proved to be unsuccessful for SmallerVGGNet. Our table contains SmallerVGGNet results for each of the listed sampling methods on a period of 200 epochs. Via repeated experimentation of training time and sampling methods, we came to the conclusion that ResNet50 at 300 epochs using oversampling provided the best metrics in almost every aspect when compared against the prior tests. After coming to this conclusion, we extended the 300 epoch ResNet50 method to the “Apathy” classification task and the results for ResNet50 in Table 1 reflect these results. Additionally notice how for each classification task, ResNet50 on image data over 300 epochs performed better across the board when compared to the similar text classification tasks executed by both support vector machine and Naive Bayes.

Based on the multi-modal fusion strategy described in Section 4.3 we were able to generate results that outperform almost every single modality experiment. We were unable to observe an increase in our precision metric for our fusion results but still maintained no lower than 0.88. In light of this, we were able to see an impressive increase in our recall measure for “Violence” which was a highly desired result. Recall represents the percentage of relevant results predicted correctly which in our case is highly beneficial.

For our multi-modal experimentation, we played around with the architecture of our fully connected layer by varying the size of our layers, the number of layers, the activation functions, kernel regularization, bias regularization as well as dropout layers. We concluded that the simple architecture outlined in Section 4.3 without dropout or layer regularization performed the best. Our “Violence” classifications in this regard required 100 epochs to reach its current metrics, while “Apathy” classification only required 40 epochs.

6 Conclusion and Future Work

In this work we present a scalable and automated implementation of polarization intent detection in an effort to aid the intent/polarization detection expert community. To this end we present a data-driven approach to this that successfully trains models on misinformation/disinformation social media ads generated by a single entity by means of single/multi-modal machine/deep learning methods. These pre-trained models can be used via transfer learning for similar intent detection tasks.

The primary direction we want to take going forward would be to extend our methodology to apply to large datasets with a vast majority of unlabeled entries. Toward this end, we will start with our proposed pre-trained models and use a semi-supervised approach, namely, co-training [5], to leverage unlabeled data. We will particularly study two datasets.

The first dataset in our sights would be a database of fact-checked content curated and published by CoronaVirusFacts Alliance. This dataset contains a multitude of misinformation/disinformation related to Coronavirus and may contain metadata that would allow our classifiers to perform intent/polarization detection. Given the sheer size of the dataset, it would also benefit greatly from a scalable solution which we have proven to be able to provide through this work with Internet Research Agency ads. In addition to this, there is also a dataset named GeoCoV19 which is a dataset of hundreds of millions of multilingual COVID-19 tweets with location information. Given the related nature of our current dataset and this GeoCoV19 dataset, it may prove to be fruitful to extend our methodologies in this area for the same reasons as for the CoronaVirusFacts dataset.

References

[1] CoronaVirusFacts Alliance. 2019. Fighting the Infodemic: The CoronaVirusFacts Alliance. data retrieved from CoronaVirusFacts Alliance https://www.poynter.org/coronavirusfactsalliance/.

[2] Stephen Harnett, Hamilton Bean. 2019. Thwarting Russian Online Disinformation: A Rhetorical Analysis/Artivicial Intelligence Pilot study. (2019).

[3] Julia Kruk, Jonah Lubin, Karan Sikka, Xiao Lin, Dan Jurafsky, Ajay Divkaran. 2019. Integrating Text and Image: Determining Multimodal Document Intent in Instagram Posts. *EMNLP'2019* (April 2019). https://web.stanford.edu/~jurafsky/document_intent.pdf

[4] Kirian Garimella, Gianmarco De Francisci Morales, Michael Maathioudakis, Aristides Gionis. 2018. Polarization on Social Media. *KDD 2018* (2018). gvrkiran.github.io/polarization/

[5] Avrim Blum, Tom Mitchel. 1998. Combining Labeled and Unlabeled Data with Co-Training. *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory* (July 1998). https://www.cs.cmu.edu/~avrim/Papers/cotrain.pdf

[6] Umair Qazi, Muhammad Imran, Ferda Ofli. 2020. GeoCoV19: A Dataset of Hundreds of Millions of Multilingual COVID-19 Tweets with Location Information. *ACM SIGSPATIAL Special* (May 2020). data retrieved from GeoCoV19 https://crisisnlp.qcri.org/covid19.

[7] Damien Pfister, Nora Murphy, Meridith Styer, Misti Yang, Purdom Lindblad, Ed Summers. 2018. About for IRAdS website. (Sept. 2018). https://mith.umd.edu/irads/data/

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition* (2016). https://ieeexplore.ieee.org/document/7780459

[9] Hermant Purohit, Guozhu Dong, Valerie L. Shalin, Krishnaprasad Thirunarayan. 2015. Intent Classification of Short-Text on Social Media. *2015 IEEE Internatioinal Conference on Smart City/SocialCom/SustainCom* (Dec. 2015). https://www.researchgate.net/publication/301932124_Intent_Classification_of_Short-Text_on_Social_Media

[10] Karen Simonyan, Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556* (Sept. 2014). https://arxiv.org/pdf/1409.1556.pdf