IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 12, No. 2, June 2023, pp. 956∼965
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i2.pp956-965

Combating propaganda texts using transfer learning

Malak Abdullah[1], **Dia Abujaber**[1], **Ahmed Al-Qarqaz**[1], **Rob Abbott**[2], **Mirsad Hadzikadic**[2]
[1]Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan
[2]Department of Computer Science, University of North Carolina at Charlotte, North Carolina, United State Amerika

Article Info

Article history:

Received Oct 19, 2021
Revised Nov 6, 2022
Accepted Dec 6, 2022

Keywords:

Bidirectioal encoder representations from transformers
Machine learning
Propaganda
Robustly optimized BERT pretraining approach
Transformers

ABSTRACT

Recently, it has been observed that people are shifting away from traditional news media sources towards trusting social networks to gather news information. Social networks have become the primary news source, although the validity and reliability of the information provided are uncertain. Memes are crucial content types that are very popular among young people and play a vital role in social media. It spreads quickly and continues to spread rapidly among people in a peer-to-peer manner rather than a prescriptive. Unfortunately, promoters and propagandists have adopted memes to indirectly manipulate public opinion and influence their attitudes using psychological and rhetorical techniques. This type of content could lead to unpleasant consequences in communities. This paper introduces an ensemble model system that resolves one of the most recent natural language processing research topics; propaganda techniques detection in texts extracted from memes. The paper also explores state-of-the-art pre-trained language models. The proposed model also uses different optimization techniques, such as data augmentation and model ensemble. It has been evaluated using a reference dataset from SemEval-2021 task 6. Our system outperforms the baseline and state-of-the-art results by achieving an F1-micro score of 0.604% on the test set.

Corresponding Author:

Malak Abdullah
Department of Computer Science, Jordan University of Science and Technology
Irbid, Jordan
Email: mabdullah@just.edu.jo

1. INTRODUCTION

Social media platforms and micro-blogging are becoming more popular than ever and have a massive impact on our daily lives. It has fundamentally changed the way people interact, communicate, think and entertain. Despite all its benefits, social media is one of the primary sources for spreading fake news and propaganda [1]. A study by Massachusetts Institute of Technology (MIT) researchers found that fake news is 70% more likely to be retweeted [2]. Information is published on social media without third-party verification, allowing people to share unverified information and reach diverse audiences. Besides fake news, propaganda has also played an essential role in changing and manipulating people's thoughts [3]. Propaganda is defined as the dissemination of rumor or information, whether true or not, used to persuade the reader to accept ideologies and prejudices to either help or hurt a particular institution, movement, group, or person [4]. It is an old term and has been in use around the world since the World Wars [5], [6]. The reasons behind propagating fake news can be financial reasons or political purposes, such as affecting elections and threatening democracies [7].

Recently, numerous researches have been motivated to identify and detect propaganda on social media sites, especially Twitter and news headlines [8]–[11]. Twitter is considered a powerful tool for shaping people's opinions and an effective channel for spreading news. For example, extremists can shape ideologies to increase the visibility of specific topics to reach a wide range of audiences. Volkova *et al.* [12] studied and explored propaganda and other types of news on Twitter about the terrorist attacks in Brussels in 2016. They developed models for detecting suspicious or verified tweets news and predicting propaganda and deception. They used publicly available platforms; PropOrNot to annotate suspicious accounts and create a list of trusted accounts. Logistic regression (LR), long short term memory (LSTM) [13], and convolutional neural network (CNN) [14] were tested with a set of features, such as DOC2VEC (representing documents as a vector), TF-IDF (term frequency-inverse document frequency), and GLoVe (global vectors for word representation) embedding for news classification. The results showed that LSTM and CNN outperformed the LR, and the addition of the linguistic features boosted performance for all models. Al-Omari *et al.* [15], applied XGBoost and several neural network architectures, such as LSTM, bidirectional LSTM, along with pre-trained transformer bidirectioal encoder representations from transformers (BERT) to classify the text into propaganda or non-propaganda.

Internet memes are among the latest developments in producing social media content. Memes are typically short captions with an image template mocking particular concepts or events, such as elections. Memes can go viral in popularity within a matter of days [16]. Therefore, it is a powerful method to be used by political and marketing campaigns for manipulation and persuasion. In other words, it is the new form of persuasion techniques. However, this type of content can be dangerous because it is unexpected for most people and gets absorbed quickly in their thoughts. Propaganda and memes are significant research areas and central topics of investigation recently. As a new topic of interest, there is a lack of work in this field; there is a small amount of open-source data available to use. Therefore, this research aims to understand how detecting persuasion techniques in texts can be automated using deep learning techniques without human intervention. Another important goal is to understand how data augmentation can help address a small data size problem. The current study experimented with pre-trained transformers on propaganda and different data augmentation techniques on the text to analyze its influence. The proposed model, PropaMemes, comprises fine-tuning pre-trained language models on propaganda technique classification of memes. It leveraged ensemble model that consists of robustly optimized BERT pretraining approach (RoBERTa) [17], BERT [18], decoding-enhanced BERT with disentangled attention (DeBERTa) [19]. Moreover, different data augmentation techniques and stacking have been experimented with, and the SemEval-2021 Task 6 dataset has been used fr evaluation. Our model shows superior results using augmentation and ensemble methods with pre-trained transformers. It achieved an F1_{micro} score of 0.604 that outperformed the state-of-the-art model, which is 0.593, on the test set.

The remainder of this paper is organized as : section 2 presents a description of the datasets used to train the models and an explanation of data augmentation techniques and data preprocessing. Section 3 the main finds of the research, including data augmentation analysis and model optimization. Finally, section 4 concludes this research and provides plans for future work.

2. RESEARCH METHOD

2.1. Datasets

The dataset used in this study has been published with a previous SemEval Competition [20]. There are two versions of the dataset; although the text is the same, the list of techniques is different from each other. In the text-based classification task, the dataset has 20 classes, and each sample can have multiple labels and captions extracted from the image. For the text-image-based task, except that the list of techniques is different (22 classes). It is worth noting that the dataset is in javascript object notation (JSON) format, with each instance in the dataset being an object composed of three fields: id, text, label. The text field is the target text we have to classify, and the label is the technique used with the corresponding text; it could be none or more. Figure 1 shows the distribution of labels at entry-level. The dataset is split into train-set of 687 samples, validation-set of 63 samples and test-set of 200 samples. Table 1 shows the distribution of the classes in training set. The dataset is challenging in terms of size and distribution. It is seen from Table 1 that the dataset is imbalanced as there are underrepresented classes, such as bandwagon and presenting irrelevant data (red herring).

Additionally, the current study is using the propaganda techniques corpus (PTC) [21], which consists of news articles from 13 propaganda and 36 non-propaganda news media outlets. The training set consisted of 371 articles and ended up with 874 samples after tokenization. The validation set consists of 75 articles and 176 samples after tokenization. This dataset has only 17 classes and does not include for smears, glittering generalities (virtue), obfuscation, intentional vagueness, confusion.

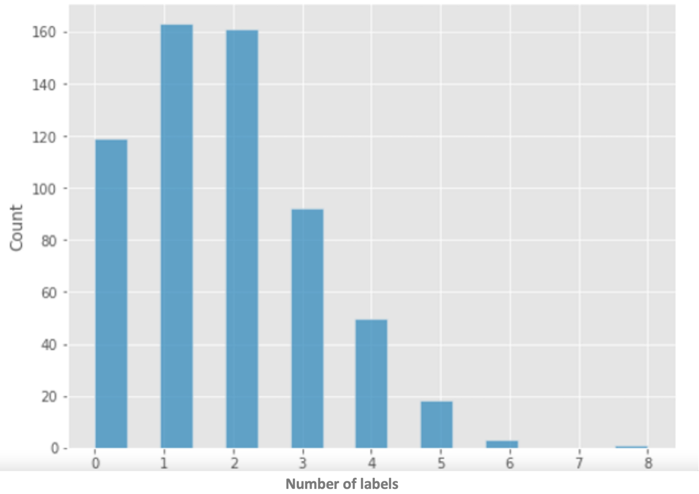

Figure 1. Distribution of labels at entry-level

Table 1. Classes distribution of training set

Technique (label)	Training set
Loaded language	313
Name-calling/labeling	188
Smears	168
Exaggeration/minimisation	52
Doubt	48
Slogans	44
Appeal to fear/prejudice	43
Whataboutism	40
Glittering generalities (virtue)	32
Casual oversimplification	27
Flag-waving	27
Misrepresentation of someone’s position (straw man)	20
Thought-terminating cliche	20
Black-and-white-fallacy/dictatorship	18
Appeal to authority	13
Reductio ad hitlerum	9
Repetition	8
Obfuscation, intentional vagueness, confusion	4
Bandwagon	2
Presenting irrelevant data (red herring)	1

2.2. Data augmentation

Data size and quality play a centric role in affecting the model performance. Collecting and annotation data could be a tedious task, and it requires domain experts to annotate the data, depending on the task complexity. To address the problem of lack of data in the propaganda domain, we have experimented with different data augmentation techniques, which is the process of producing additional data to increase the dataset size and overcome the overfitting problem [22]. Data augmentation is frequently used in computer vision [22], which includes variant of techniques, including cropping, rotation, zooming, and flipping. However, there had been previous efforts in using data augmentation techniques with text [23]. In this research, we will be using techniques presented in [23], which includes synonym replacement, random insertion, random swap, and random deletion. Table 2 shows an example of each of the augmentation techniques. In addition to these techniques, we will also experiment with back-translation. For each sample in the dataset, we have performed one or more of the following:

- Synonym replacement (SR): randomly choose n words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.

- Random insertion (RI): find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this n times.
- Random swap (RS): randomly choose two words in the sentence and swap their positions. Do this n times.
- Random d (RD): randomly remove each word in the sentence with probability p.
- Back translation (BT): translate the text into some other language then re-translate it back to the original language.

Table 2. Sample text generated from each data augmentation techniques

Technique	Output
Original text	This paper will describe our system in detecting propaganda in memes
Synonym replacement	This theme will describe our arrangement in detecting propaganda in memes
Random insertion	This key out paper will describe our system meme in detecting propaganda in memes
Random swap	This paper in describe our system in detecting propaganda will memes
Random deletion	This paper will describe system in detecting propaganda in memes
Back translation	This document describes our memorandum propaganda detection system

2.3. Data preprocessing

Cleaning and preprocessing the data is a crucial step to reduce potential noise before training any model. Unlike tweets, memes are less noisy (i.e., no hashtags, and URLs). Although the uppercase text could indicate solid emotions in typical cases, memes caption is usually written with all uppercase, which makes it unhelpful for this task; thus, we converted all the text into lowercase. We have removed numbers and special characters as they don't hold semantic information. Then we applied each of the previously described augmentation techniques on the samples with a probability of 0.2. We tested each technique's models to observe how each technique influences the model. Then we combined all the techniques into one dataset and tested the model on it. For each technique, three samples were generated without replacement (did not include the original text in the new synthesized dataset).

2.4. Metrics

The evaluation metrics for this work are Micro-F_1 and Macro-F_1, hence the dataset is extremely imbalanced. These metrics produce values that range from 0-1, where 0 refers to the lowest result and 1 to the highest result. F_1-score is the harmonic mean of the precision and recall. In other words, it gives us a compromise value between precision and recall. Macro-F_1 is calculated by taking the mean F_1-score of each label thus it puts bigger penalization when the model doesn't perform well with minority classes. Micro-F_1metric puts more emphasis on the most populated classes.

2.5. Pre-trained models (transformers)

Pre-trained language transformers, such as BERT [18], have proved to achieve high performance with a small amount of data from a typically few numbers of epochs. Recently, pre-trained transformers have gained broad popularity and adaption by the natural language processing (NLP) community. Two of the most known transformers are BERT [18] and RoBERTa [17]. Both transformers have two versions: a base version and a large version. The difference between the two versions for each transformer is the model architecture size. The base models of both BERT and RoBERTa have 12 layers of transformers block with 768 hidden size and 12 self-attention heads. The BERT large and RoBERTa large have 24 layers of transformers block with 1024 hidden size and 16 self-attention heads.

3. TRAINING AND EXPERIMENTS

This section explores the performance of BERT and RoBERTa with different data augmentation techniques combination to understand and analyze the influence of each technique. It is worth noting that the same hyper-parameters have been used in all experiments. A learning rate of 2e-5 has been used. Hence memes text is typically short; a sequence length of 100 and batch size of 8 were used. Then the model was fine-tuned for 8 epochs with AdamW optimizer. As the transformers fine-tuning is non-deterministic (i.e., each run leads to a different but close performance), three models were trained for each experiment in this study. The average was taken of the three with the standard deviation.

On the validation set, the Original dataset was used as a baseline to measure the improvement of the augmentation techniques. In most cases, the data augmentation techniques outperformed the original dataset and achieved a global $f1_{micro}$ average of 5.52% improvement and global $f1_{macro}$ average of 3.06% improvement on the validation set. An observation to make is that models performed generally better when different techniques are combined together. On the contrary, some combinations have dropped the performance. It is worth noting that this could be due to the randomness of transformers' outcomes of different training runs.

On the test set, all models trained on the data augmentation techniques outperformed the baseline (Table 3). The global $F1_{micro}$ average improvement rate is 5.42% and global $f1_{macro}$ average improvement rate is 3.06%. Similar to validation score, the best scores were achieved with combinational techniques (8 and 9). $\text{BERT}_{\text{large}}$, $\text{BERT}_{\text{base}}$ and $\text{RoBERTa}_{\text{base}}$ achieved their highest score with technique 8 and $f1_{micro}$ improvement of 6.4%, 9.6% and 5.0% respectively. $\text{RoBERTa}_{\text{large}}$ achieved the best score out of all models on technique 9 with $f1_{micro}$ improvement of 4.7%. Table 4 outlines a summary statistics of the performance on the test set and validation set.

Table 3. Data augmentation results on test set

#	Data augmentation technique	Metric	$\text{BERT}_{\text{large}}$	$\text{BERT}_{\text{base}}$	$\text{RoBERTa}_{\text{large}}$	$\text{RoBERTa}_{\text{base}}$
1	Original dataset (baseline)	$F1_{micro}$	0.470 ± 0.016	0.407 ± 0.010	0.512 ± 0.007	0.470 ± 0.001
		$F1_{macro}$	0.133 ± 0.003	0.115 ± 0.003	0.150 ± 0.006	0.129 ± 0.003
2	SR	$F1_{micro}$	0.493 ± 0.010	0.438 ± 0.014	0.550 ± 0.014	0.486 ± 0.001
		$F1_{macro}$	0.148 ± 0.002	0.125 ± 0.003	0.220 ± 0.018	0.136 ± 0.001
3	RI	$F1_{micro}$	0.496 ± 0.009	0.458 ± 0.003	0.534 ± 0.010	0.475 ± 0.013
		$F1_{macro}$	0.145 ± 0.005	0.130 ± 0.001	0.200 ± 0.019	0.132 ± 0.003
4	RS	$F1_{micro}$	0.480 ± 0.002	0.453 ± 0.005	0.532 ± 0.012	0.486 ± 0.011
		$F1_{macro}$	0.149 ± 0.004	0.128 ± 0.001	0.190 ± 0.020	0.137 ± 0.002
5	RD	$F1_{micro}$	0.493 ± 0.019	0.449 ± 0.011	0.546 + 0.003	0.481 ± 0.013
		$F1_{macro}$	0.141 ± 0.009	0.125 ± 0.006	0.200 ± 0.009	0.135 ± 0.002
6	SR+RI+RS+RD	$F1_{micro}$	0.493 ± 0.015	0.461 ± 0.007	0.542 ± 0.004	0.490 ± 0.010
		$F1_{macro}$	0.180 ± 0.012	0.137 ± 0.009	0.224 ± 0.018	0.182 ± 0.021
7	BT	$F1_{micro}$	0.534 ± 0.005	0.474 ± 0.004	0.556 ± 0.011	0.497 ± 0.010
		$F1_{macro}$	0.190 ± 0.014	0.135 ± 0.001	0.239 ± 0.016	0.156 ± 0.008
8	All techniques (excluding original samples)	$F1_{micro}$	**0.531 ± 0.004**	**0.503 ± 0.010**	0.550 ± 0.013	**0.520 ± 0.013**
		$F1_{macro}$	0.208 ± 0.031	0.185 ± 0.019	0.246 ± 0.014	0.193 ± 0.008
9	All techniques (including original samples)	$F1_{micro}$	0.529 ± 0.008	0.496 ± 0.001	**0.559 ± 0.006**	0.514 ± 0.020
		$F1_{macro}$	0.237 ± 0.005	0.190 ± 0.006	0.245 ± 0.022	0.192 ± 0.026

Table 4. Summary results of each model on test and validation sets. max micro/macro score is the best score achieved on one or a combination of the data augmentation techniques

	Validation set						Test set					
	Baseline	$F1_{micro}$ Max Score	Average Improve	Baseline	$F1_{macro}$ Max Score	Average Improve	Baseline	$F1_{micro}$ Max Score	Average Improve	Baseline	$F1_{macro}$ Max Score	Average Improve
$\text{BERT}_{\text{large}}$	0.494	0.578	+6.05%	0.234	0.343	+8.00%	0.470	0.534	+3.61%	0.133	0.237	+4.17%
$\text{BERT}_{\text{base}}$	0.427	0.566	+8.05%	0.214	0.331	+5.40%	0.407	0.503	+5.95%	0.115	0.190	+2.93%
$\text{RoBERTa}_{\text{large}}$	0.568	0.580	-0.03%	0.298	0.371	+4.50%	0.512	0.556	+3.41%	0.150	0.246	+7.05%
$\text{RoBERTa}_{\text{base}}$	0.469	0.569	+6.02%	0.222	0.339	+7.60%	0.470	0.520	+2.36%	0.129	0.193	+2.88%

3.1. Performance optimization using PTC corpus

Due to the incompatibility of class number and text nature between PTC corpus [21] and memes data-set, the training process was divided into two stages. In the first stage, the model was fine-tuned on PTC corpus. In the second stage, since the corpus consists only of 17 classes, we discarded the classification layer after fine-tuning the model and replaced it with another classification layer of size 20 (the number of classes we originally have) fine-tuned on the memes data-set. Figure 2 illustrates the training process stages, and Table 5 shows the hyper-parameters used with each stage (the epoch with the best performance was saved and selected). $\text{RoBERTa}_{\text{large}}$ model was trained on that process with memes dataset from technique 9 (Table 3). Then, the other models $\text{RoBERTa}_{\text{large}}$, $\text{BERT}_{\text{large}}$, $\text{DeBERTa}_{\text{large}}$ were trained on the described training process. Table 6 shows the performance achieved for each model. There is an obvious improvement on the validation set performance with $\text{BERT}_{\text{large}}$; however, its performance did not improve much on the test set.

On the other side, RoBERTa$_{\text{large}}$ performance has dropped with the validation but improved on the test set. Although, DeBERTa$_{\text{large}}$ was not experimented with at the data augmentation analysis section, it has achieved a remarkable performance which outperformed both of RoBERTa$_{\text{large}}$ and BERT$_{\text{large}}$ and closest to the current state-of-the-art with a difference of 0.009 f1$_{\text{micro}}$.

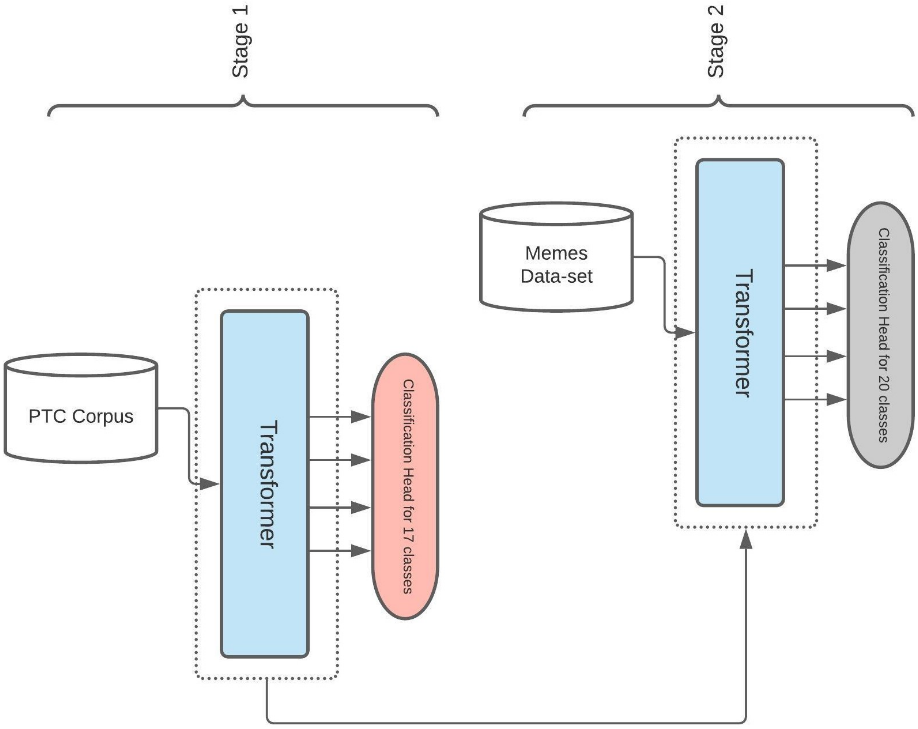
Stage 1
Stage 2
Memes Data-set
Transformer
Classification Head for 20 classes
PTC Corpus
Transformer
Classification Head for 17 classes

Figure 2. Training stages illustration

Table 5. Hyper-parameters used at the optimization step with the 2 stages training process

Hyper-parameters	Stage 1	Stage 2
Learning-rate	2e-05	3e-05
Max sequence length	150	100
Batch size	8	16
Number of epochs	8	8
Optimizer	AdamW	AdamW
Loss	BCE	BCE

Table 6. Models performance on 2-stage training process

	Validation set		Test set	
Model	F1micro	F1macro	F1micro	F1macro
Debertalarge	0.621	0.409	0.584	0.285
BERTlarge	0.620	0.358	0.538	0.183
RoBERTalarge	0.535	0.298	0.575	0.222

3.2. Performance optimization using ensemble learning

Ensemble learning combines the output of multiple base-learners to provide a final prediction. The main advantage of ensemble learning is the reduction of variance and prediction improvement. There are different methods for a model ensemble, such as stacking and averaging. This study used ensemble learning to compose the final model by combining the output of multiple base-learners to provide a final prediction.

One of the most intuitive and straightforward methods is averaging that averages the output of each model at prediction time. The model achieved a macro score of 0.346 and a micro score of 0.588 with a threshold of 0.5 on the validation set using averaging ensemble technique. We experimented with other thresholds, the lower the threshold, the better the score is. The best score was accomplished by a threshold of 0.3. Running the ensemble model with a threshold of 0.3 on the test set attained a micro score of 0.6 and a macro score of 0.272 on the test set, which outperforms the current state-of-the-art with less ensembled models.

Another ensemble method is stacking, which involves two levels in the classification process. Level-0 consists of base models that are trained on the training set, and their output is passed to the next level. Level-1 consists of meta-models that train on the output of the base models from out-of-sample data that wasn't used in training the base classifiers, and the output with the ground truth labels are used to train the meta classifiers. We used the same data samples on both base classifiers and meta classifiers as we have a small dataset. Experimenting with the original process gave poor results since when we split the training set, the base classifiers trained on fewer data, thus degrading performance. The meta-classifiers weren't able to compensate for that loss. We used scikit-multilearn library to train the meta-models. We experimented with support vector classification (SVC) and LR for stacking. We also used the logits output from the transformers to train the meta-models. As LR and SVC are a binary classifiers we used OneVsRestClassifier and chain classifiers approaches to transform multi-label problem into a individual single-label problems. In OneVsRestClassifier, the problem is decomposed into multiple single-label problems by training a model for each label from the downstream logits. One of the issues of that approach is that it doesn't take the correlation of classes into account. We experimented with our transformers and got a micro score of 0.591 and a macro score of 0.265 on the test set with SVC. The other approach is chain classifiers; the main advantage of this approach over the OneVsRestClassifier is that the correlation of classes is involved in the classification model. This approach creates N classifiers where N is the number of classes and the output of classifiers n_i is the input for classifiers n_j such that i ¡ j. It achieved a micro score of 0.604 and a macro score of 0.289 on the test with SVC. Table 7 summaries the ensemble model experiments.

Table 7. Ensemble models performance with SVC, logistic regression and weighted average

		Validation set		Test set	
		$F1_{micro}$	$F1_{macro}$	$F1_{micro}$	$F1_{macro}$
SVC	OneVsRestClassifier	0.604	0.358	0.591	0.265
	ClassifierChain	0.609	0.216	0.604	0.289
Logistic Regression	OneVsRestClassifier	0.605	0.215	0.598	0.265
	ClassifierChain	0.605	0.365	0.601	0.265
Average with threshold = 0.3		0.635	0.389	0.600	0.272

3.3. Comparison and evaluation

This subsection shows a comparison between the results of our proposed model, PropaMemes, and other models' results from the official shared task leaderboard [20]. Table 8 summarizes the comparison. The baseline model was provided by the organizers, which created random labels [20].

The current top performance is achieved by [24]. Their solution is an ensemble model system that ensemble five Transformers (BERT, RoBERTa, XLNET, DeBERTa, and ALBERT) by taking the average probability output of each. They have used focal loss (FL) [25] as it gives lower weights for easy examples and puts more emphasis on hard examples to reduce the effect of the imbalanced dataset problem. They first fine-tuned their transformers on the PTC corpus. Then they continued training the models with memes dataset. There final ensemble model has achieved $F1_{micro}$ of 0.593 and $F1_{macro}$ of 0.289 on the test set. Our proposed model (PropaMeme) achieved an $F1_{micro}$ score of 0.604 on the test set using model ensembles three pre-trained transformers of Bert, RoBERTa, and DeBERTa with output averaging. Figure3 shows the architecture of the model.

Table 8. A comparison between highest ranked models

Model	F1-micro
BaseLine	0.064
MinD	0.593
PropaMemes	**0.604**

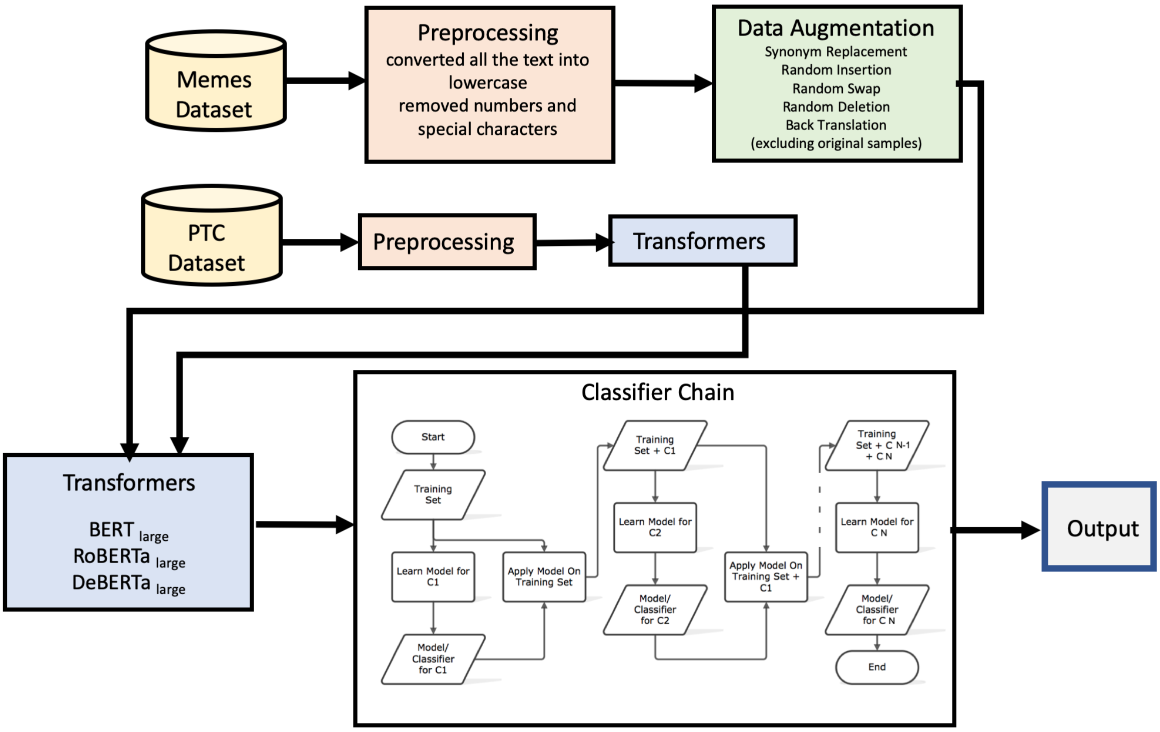

Figure 3. PropaMemes model architecture

4. CONCLUSION

In this paper, we have experimented with several data augmentation techniques for text and it turned out that they did improve the results almost in every case. We have also used and external articles dataset to improve the performance of our model. Moreover, we explored some stacking and model ensemble methods to further improve the model performance. Finally, we were able to achieve a $F1_{\text{micro}}$ of 60% on test set using model ensemble of Bert, RoBERTa and DeBERTa with output averaging. Our future work will focus on incorporating images along with text in making inference as some of samples hold meaning on their visual content. We will also work on further improving the score of the presented system with a fewer ensembled models. Although we tried to reduce the affect of small dataset size using data augmentation, but having a diverse balanced dataset is still essential. We will try to collect and annotate more samples using crowd-sourcing and semi-supervised learning.

REFERENCES

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media: a data mining perspective,” *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017. doi: 10.1145/3137597.3137600.

[2] S. Vosoughi, D. Roy, and S. Aral, “The spread of true and false news online,” *Science*, vol. 359, no. 6380, pp. 1146–1151, Mar. 2018, doi: 10.1126/science.aap9559.

[3] A. Gelfert, “Fake news: A definition,” *Informal Logic*, vol. 38, no. 1, pp. 84–117, 2018, doi: 10.22329/il.v38i1.5068.

[4] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, “FakeNewsNet: a data repository with news content, social context, and spatiotemporal information for studying fake news on social media,” *Big Data*, vol. 8, no. 3, pp. 171–188, 2020, doi: 10.1089/big.2020.0062.

[5] E. W. Fellows, “Propaganda: history of a word,” *American Speech*, vol. 34, no. 3, pp. 182–189, Oct. 1959, doi: 10.2307/454039.

[6] Garth S. Jowett and Victoria O’Donnell, “What is propaganda, and how does it differ from persuasion?,” *Propaganda and Persuasion*, vol. 1–48, 2012.

[7] C. Shao, G. L. Ciampaglia, O. Varol, K. C. Yang, A. Flammini, and F. Menczer, “The spread of low-credibility content by social bots,” *Nature Communications*, vol. 9, no. 1, pp. 1–9, 2018, doi: 10.1038/s41467-018-06930-7.

[8] A. Barrón-Cedeño, I. Jaradat, G. Da San Martino, and P. Nakov, “Proppy: organizing the news based on

their propagandistic content," *Information Processing and Management*, vol. 56, no. 5, pp. 1849–1864, 2019, doi: 10.1016/j.ipm.2019.03.005.

[9] O. Altiti, M. Abdullah, and R. Obiedat, "JUST at SemEval-2020 task 11: detecting propaganda techniques using BERT pretrained model," in *Proceedings of the 14th International Workshop on Semantic Evaluation, SemEval*, 2020, pp. 1749–1755, doi: 10.18653/v1/2020.semeval-1.229.

[10] D. Abujaber, A. Qarqaz, and M. A. Abdullah, "LeCun at SemEval-2021 task 6: detecting persuasion techniques in text using ensembled pretrained transformers and data augmentation," in *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, 2021, pp. 1068–1074, doi: 10.18653/v1/2021.semeval-1.148.

[11] M. Abdullah, O. Altiti, and R. Obiedat, "Detecting propaganda techniques in english news articles using pre-trained transformers," in *2022 13th International Conference on Information and Communication Systems, ICICS*, 2022, pp. 301–308, doi: 10.1109/ICICS55353.2022.9811117.

[12] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, "Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, vol. 2, pp. 647–653, doi: 10.18653/v1/P17-2102.

[13] S. Hochreiter and J. U. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, vol. 86, no. 11, pp. 2278–2324, doi: 10.1109/5.726791.

[15] H. Al-Omari, M. Abdullah, O. AlTiti, and S. Shaikh, "JUSTDeep at NLP4IF 2019 task 1: propaganda detection using ensemble deep learning models," in *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, 2019, pp. 113–118, doi: 10.18653/v1/D19-5016.

[16] A. Kaltenhauser, N. Terzimehić, and A. Butz, "Memeography: understanding users through internet memes," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, May 2021, pp. 1–7, doi: 10.1145/3411763.3451581.

[17] Y. Liu et al., "RoBERTa: a robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019, doi: 10.48550/arXiv.1907.11692.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, Oct. 2018, doi: 10.48550/arXiv.1810.04805.

[19] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: decoding-enhanced BERT with disentangled attention," *arXiv preprint arXiv:2006.03654*, 2020, doi: 10.48550/arXiv.2006.03654.

[20] D. Dimitrov et al., "SemEval-2021 task 6: detection of persuasion techniques in texts and images," *arXiv preprint arXiv:2105.09284*, Apr. 2021, doi: 10.48550/arXiv.2105.09284.

[21] G. da San Martino, A. Barrón-Cedeño, H. Wachsmuth, R. Petrov, and P. Nakov, "SemEval-2020 task 11: detection of propaganda techniques in news articles," in *Proceedings of the 14th International Workshop on Semantic Evaluation, SemEval*, 2020, pp. 1377–1414, doi: 10.18653/v1/2020.semeval-1.186.

[22] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019, doi: 10.1186/s40537-019-0197-0.

[23] J. Wei and Z. Kai, "EDA: easy data augmentation techniques for boosting performance on text classification tasks," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, 2020, pp. 6382–6388.

[24] J. Tian, M. Gui, C. Li, M. Yan, and W. Xiao, "MinD at SemEval-2021 task 6: propaganda detection using transfer learning and multimodal fusion," in *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, 2021, pp. 1082–1087, doi: 10.18653/v1/2021.semeval-1.150.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

BIOGRAPHIES OF AUTHORS

Malak Abdullah Received her Ph.D. degree in Computer Science from the University of North Carolina at Charlotte, United State Amerika, in 2018. She is currently an Assistant Professor at Jordan University of Science and Technology, working in the Department of Computer Science. Her research interests include data science, natural language processing, artificial intelligence, machine and deep learning. She can be contacted at email: mabdullah@just.edu.jo.

Dia Abu-Jaber Received his B.S degree in computer science from Jordan University of Science and Technology, Jordan, in 2021. He is currently working as Software Engineering at Amazon. His research interests include natural language processing (NLP), deep learning and machine learning. He has co-authored several technical papers in established conferences in fields related to NLP. He can be contacted at email:diaa1999abujaber@gmail.com.

Ahmed Al-Qarqaz Received his B.S degree in Computer Science from Jordan University of Science and Technology, Jordan, in 2022 and he is currently a master student in Computer Science. His research interests include natural language processing (NLP), dep learning and machine learning. He has co-authored several technical papers in established conferences in fields related to NLP. He can be contacted at email: ahmedqarqaz6@gmail.com.

Rob Abbott Received his Ph.D. degree in Computer Science from the University of North Carolina at Charlotte, United State Amerika, in 2021. He is the the Principal Architect/Chief Data Officer at Project Indigo and also the Founder at Abbottics. He can be contacted at email: rob@abbottics.ai.

Mirsad Hadzikadic is in the Software and Information Systems Department and School of Data Science at UNC Charlotte. He is the Director of the Complex Systems Institute at the same university. He was the founding Dean of UNC Charlotte's College of Computing and Informatics, founding Executive Director of Data Science Initiative, and Chair of the Department of Computer Science. Dr. Hadzikadic's research is centered on complex adaptive systems, simulation and modeling, health informatics, population health, data mining/analytics, and cognitive science/neuroeconomics. He can be contacted at email: mirsad@uncc.edu.