False Article Detection with Weakly Supervised Learning

Bhupender kumar Saini[1] , **Swathi Chidipothu Hare**[2] , **Ravineesh Goud**[3] , **Mayur Rameshbhai Waghela**[4] , **Lokesh Sharma**[5] , **and Ipek Baris**[6]

University of Koblenz-Landau, Campus Koblenz

{bksaini,swathihare,ravineeshgoud,mayurwaghela,lokesharma,ibaris}@uni-koblenz.de

Abstract

The creation, accessibility and consumption of fake news is ubiquitous in this era with the rise of social media platforms and internet enabled devices. The news shape the opinion of readers, society, and there is a high chance for the users to be stuck in a loop of "legit or fake" confusion. For supervising techniques, it become a challenge to consistently present true information to readers/viewers, as it requires large amount of training examples with ground-truth labels which is quite hard to collect considering in real scenario. And, achieving equal or greater performance in supervised learning in lack of labelled data is another challenge in detecting fake news. In this research paper, we have designed weakly supervised learning models based on *mean teacher*, *virtual adversarial training*, and *pseudo labelling* by introducing 3 different noise generation techniques along with adding perturbations in the embedding layer to detect authenticity of news articles. We also presented a comparison of these models and proposed a new line of future work.

Keywords: Fake news, Weakly supervised, Mean Teacher, Virtual Adversarial Training, Noise, Dropouts, Regularization, Adversarial training, Pseudo Labelling.

1 Introduction

Fake news is deliberately presenting false or misleading claims as news, where the claims are misleading by design. The phrase "by design" is then explicated in terms of systemic features of the process of news production and dissemination [1]. The means of communications have evolved over the years and so does the idea of news consumption. The news reader now has a wide variety of alternatives a click away in this age of digital epoch.

The spread of propaganda and misinformation is now ubiquitous over online space and has raised questions on the credibility of social media, news media platforms, and news authors. When original events are twisted by misinformation in a subsequent news article, people are more likely to recognize the false information as the original event data and less likely to identify the correct facts [2]. An ample amount of conversations, researches are centered around the implications of fake news in politics after the 2016 US Presidential elections but the fake news is now deepening its roots into other spectrum and now has started to dominate other elements of society.

In the current scenario where the dynamics of the world are changing rapidly, fake news has managed to contribute in hampering the wider economy, causing tension amongst the societies, communities and it would be dangerous to imagine what kind of potential fake news possesses to cause chaos in unprecedented events. The detection of fake news is becoming more and more challenging as the large amount of false information is spreading quickly through social media. To tackle this problem, and to handle the enormous amount of data at same time, deep learning models have already established their identity and shown prominent result in the detection of fake news. Supervised learning techniques have achieved great success when there is strong supervision information like a large amount of training examples with ground-truth labels. In real tasks, however, collecting supervision information requires costs, and thus, it is usually desirable to be able to do weakly supervised learning [3].

2 Background and Related Work

As with the increase in complexity of the information ecosystem, the usage of the term 'fake news' suppress crucial distinctions, while the information disorder comes under many flavors. Information disorder is categorized under three major types i.e. *misinformation*, *disinformation*, and *malinformation* [4]. Misinformation can be simply defined as false, mistaken, or misleading information, 'disinformation' entails the distribution, assertion, or dissemination of false, mistaken, or misleading information in an intentional, deliberate, or purposeful effort to mislead, deceive, or confuse. Malinformation is to describe genuine information that is shared with an intent to cause harm.

Dong-Hyun Lee [5], proposed the approach of training model with labeled and unlabeled data simultaneously. In which unlabeled data is assigned to the class having the maximum predicted probability. On applying de-noising auto-encoder and dropout [6] to their model, it outperformed CNN methods for semi-supervised learning. The main take away is

that weak labels with regularization techniques like introducing noise, dropout can produce better result.

Another proposed technique is WeFEND [7], a reinforced weakly supervised fake news detection framework. Their framework consists of three main components: the annotator, the reinforced selector and the fake news detector. Annotator is responsible for automatically assignment of weak labels to unlabeled news based on user's report. Reinforced selector uses reinforcement learning techniques to choose high quality samples from weakly labeled data set and filters out the low-quality data. The fake news detector identifies fake news based on the news content. Their proposed WeFEND model performed well compared to other state-of-art methods like Hybrid Deep Model [8], CNN or LSTM in supervised, semi-supervised and weakly supervised fashion. The WeFEND model still rely on the user's report in order to label the unlabeled data using Annotator.

Tarvainen, Antti and Valpola, Harri [9] proposed mean teacher approach, where it assumes a model in a dual role as a teacher and a student. First, it learns as a student then as a teacher generates target which will be for learning as a student. They claimed that teacher model perform better and more robust in contrast to student model. However, both the model can be used for prediction. This approach performed well in speech recognition and image processing tasks.

2.1 Proposed approach

To achieve the fake news detection using weakly supervised approaches, we taken the algorithms that already have shown state-of-art results in the area of computer vision. Our main goal is to implement and evaluate their performance under Natural language processing domain. Additionally, observing the performance of models by introducing different noise strategies and weakly supervised learning with baseline models.

In this paper, we have introduced 3 different noise strategies (i.e. synonym replacement, dropping words, synonym and dropout) and 1 approach of utilizing unlabelled data. Their implementation during the training of our model, and further adding perturbations in the embedding layer of the language model by using weights from pre-trained models. In addition, we have inspired from the tri-training approach as mentioned in [10] and designed an bi-training approach for pseudo label which has performed better than supervised approach and requires 10% of labeled data for training.

We have used Supervised BiLSTM and Label propagation as baseline models and further implemented 4 different variants of Mean Teacher, VAT, and pseudo labelling. The overview of our approach is shown in Figure 1. We have chosen accuracy, precision, recall, f1 score, binary cross entropy cost, and ROC curve as our evaluation parameters. Our findings shows that, in terms of test-accuracy, precision-true, precision-fake, recall-true, f1-true, and f1-fake, Pseudo Label has outperformed other models (0.731 ± 0.01,0.718 ± 0.03, 0.756 ± 0.02, 0.797 ± 0.04, 0.753 ± 0.02, 0.698 ± 0.03) respectively. Mean teacher combining with synonym replacement noise technique has shown lowest binary loss i.e. 0.552 ± 0.03. VAT model has outperform other models with value of 0.662 ± 0.07 for recall-fake. Further, the ROC curve shows that pseudo label model has shown promising result in fake news detection, however as per the evaluation, the VAT model has also shown consistent performance considering all the evaluation parameters.

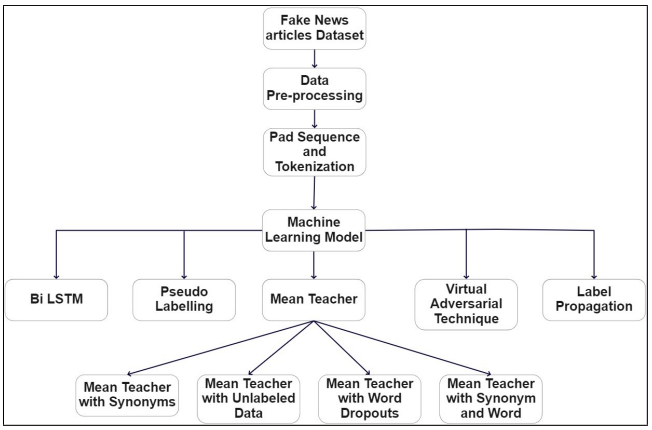

Figure 1: Flow chart to illustrate the overview

3 Methodology

3.1 Mean Teacher

In the mean teacher model, two identical models are trained with two different strategies called student and teacher model. In which, only student model is trained, however, during training exponential moving weights are assigned to the teacher hence it is called as Mean teacher. As shown in Figure 2, two cost function plays important role while back-propagating i.e. classification cost and consistency cost. Classification cost($C(\theta)$) is calculated as binary cross entropy between label predicted by student model and original label. Consistency cost($J(\theta)$) is mean squared difference between the predicted outputs of student (weights θ and noise η) and teacher model (weights $\hat{\theta}$ and noise η'). The mathematical declaration is as follows [9].

$$J(\theta) = \mathbb{E}_{x,\eta,\eta'}[\|f(x,\theta,\eta) - f(x,\hat{\theta},\eta')\|^2] \tag{1}$$

While back propagating in student model, the overall cost ($O(\theta)$) is calculated with given formula

$$O(\theta) = rC(\theta) + (1-r)J(\theta) \tag{2}$$

During training, exponential moving average(EMA) weights of the student model are assigned to the teacher model at every steps and the proportion of weights assigned is controlled by parameter alpha(α). As mentioned in equation 3, while assigning weights, teacher model holds its previous weights in alpha(α) proportion and $(1-\alpha)$ portion of student weights.

$$\hat{\theta}_t = \alpha\hat{\theta}_{t-1} + (1-\alpha)\theta_t \tag{3}$$

As per the claim by Antti Tarvainen et al. [9], after a particular epoch during training of teacher model, it starts performing better than the student model in terms of test accuracy, precision, and losses. However, tuning of hyper-parameters

alpha(α) and ratio(r) is required to get the expected result. One of the important factors that plays crucial role in adding robustness in the model is the introduction of noise during training. Noise is one of the best techniques of regularization [6]. In our proposed research, the study on different noise strategies in the mean teacher model and achieving weakly supervised learning is our main focus.To achieve weakly supervised learning we proposed two methods of utilizing unlabeled data as follows:

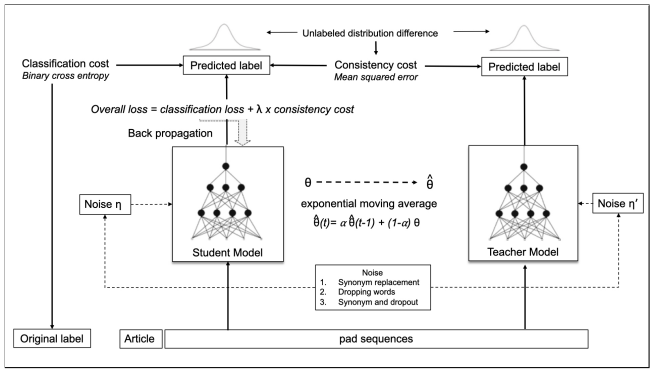

Figure 2: Mean teacher model with different methods of noise

1. The mean squared difference between student and teacher model predicts output distribution of unlabeled data as consistency cost $J(\theta)$ during training student model as shown in Figure 2. It's assumed that unlabeled data will be having true distribution same as label data [11]. By adding distribution differences while training, the model tries to reduce the difference between student and teacher output distribution.
2. To find out the most similar words under same vocabulary domain in our case politics. We have achieved this by creating embedding of label and unlabeled datasets. Adding unlabeled data increases the number of words and also provides enough new words to replace in labeled data which increases syntactic accuracy during adversarial training. In this case, as consistency cost between the student and teacher model, we are calculating mean squared error between the predicted output of both student and teacher model.

Noise generation strategies

For introducing noise in our study, we implemented three strategies as follows:

1. *Synonym replacement*: Replacing articles words with their synonym(*most similar word*). To achieve this, we trained FastText[1] embedding model provided by gensim[2] to find the most similar words. FastText embeddings are trained for understanding the morphological structure and it perform better than word2vec[3] in syntactic tasks. In the proposed approach, the example of FastText similarity model is shown in Figure 3. During training, first, we decide with probability(*p1*) that the article should be considered for replacement or not. Once selected, with second probability(*p2*), we decide word should change or not.

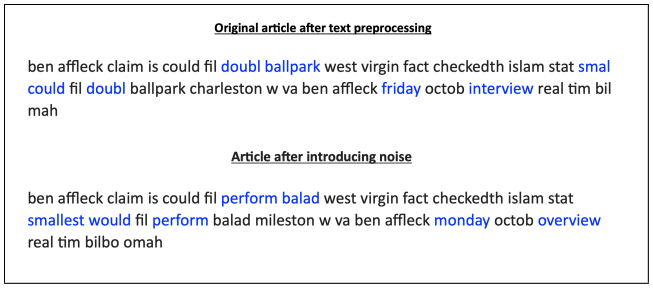

Figure 3: Adversarial noise example

2. *Dropping words*: The approach of dropping words is based on probability($\rho 1$), we decided that word should be dropped or kept as it is by replacing particular word to zero. Then while calculating consistency cost, we calculated MSE (mean squared error) with different dropouts in student and teacher output as shown in Figure 2. As in this approach, we have not used unlabeled data, so this approach is considered as supervised. However, using dropout only in student model and unlabeled data distribution difference as consistency cost can be a scope for future research.
3. *Synonym and Dropout* : In this strategy, we first replaced the words with most similar words using probabilities(*p1* and *p2*) and then drop the words with probabilities($\rho 1$).

Experimental setting

For this, we have chosen our model architecture as BiLSTM(Bi-directional Long Short Term Memory) model for student and teacher as shown in Figure 4. We tried to keep our model simple with 5 layers and activation function only at the outer layer i.e. Sigmoid activation function and 100 maximum length for input layer. For synonym noise strategy, we are using 3269 unlabelled data while creating embedding and vocabulary. However, for unlabeled data strategy, we are considering 600 unlabeled data to calculate consistency cost during training due to computational system and time constraint, however considering as much as possible unlabeled data is highly recommended. In the Mean Teacher approach, we train student model by normal backpropagation using adam optimizer with learning rate 0.0001, whereas the teacher model's weights are updated using the exponential moving average(EMA). In EMA, we keep some portion(alpha α) of old Teacher's weights and add $(1-\alpha)$ portion of new student's updated weights every step as mentioned in the algorithm 1. The value of alpha (α) and ratio (r) is set to 0.99 and 0.5 respectively during training. Further details of parameters are shown in table 1. Implementation code for the same is available at Github repository [4].

[1]https://fasttext.cc/

[2]https://radimrehurek.com/gensim/

[3]https://radimrehurek.com/gensim/models/word2vec.html

[4]https://github.com/bksaini078/fake_news_detection

Algorithm 1: Mean Teacher Algorithm

Data: train set $(\mathcal{X}, \mathcal{Y})$, Unlabel data($\mathcal{Z}$)
Hyper parameters: r, α, $p1$, $p2$, $\rho 1$, $\rho 2$, *epochs*;
Create Model : *student*(θ), *teacher*($\hat{\theta}$) ;
Train *teacher* for 1 epoch;
while *epochs* **do**
  while *steps* **do**
    1: Insert noise with probability as per strategies ($p1$, $p2$, $\rho 1$, $\rho 2$) in $\mathcal{X}$ i.e. $\mathcal{X}_\eta$;
    2: *student*($\mathcal{X}_\eta$) = $\mathcal{Y}_\eta$;
    3: Classification cost ($C(\theta)$) =Binary Cross Entropy($\mathcal{Y}$,$\mathcal{Y}_\eta$);
    4: Again create different noise data as mentioned in step 1 i.e. $\mathcal{X}_{\eta'}$;
    5: *teacher*($\mathcal{X}_{\eta'}$) = $\mathcal{Y}_{\eta'}$;
    6: Calculate Consistency cost $J(\theta)$=Mean Squared Error($\mathcal{Y}_\eta$,$\mathcal{Y}_{\eta'}$);
    7: Calculate Overall cost $O(\theta) = rC(\theta) + (1-r)J(\theta)$;
    8: Calculate *gradients*, $O(\theta)$ w.r.t θ ;
    9: Apply *gradients* to θ;
    10: Update Exponential Moving average of θ to $\hat{\theta}$ i.e. $\hat{\theta}_t = \alpha\hat{\theta}_{t-1} + (1-\alpha)\theta_t$;
  end
end

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 100)]             0
_________________________________________________________________
embedding (Embedding)        (None, 100, 128)          38400
_________________________________________________________________
bidirectional (Bidirectional (None, 256)               263168
_________________________________________________________________
dense (Dense)                (None, 2)                 514
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 3
=================================================================
Total params: 302,085
Trainable params: 302,085
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: Model summary for Student and Teacher

Parameter Name	Value
K-fold	10
Epoch	15
Learning rate *lr*	0.0001
Optimizer	adam
Batch size	64
Alpha α	0.99
Ratio *r*	0.50
Probability *p1*	0.5, syn and drop-0.4
Probability *p2*	0.3, syn and drop-0.2
Probability $\rho 1$	0.2, syn and drop-0.2
Test/Train split	10%/90%

Table 1: Parameters detail for mean teacher, *syn and drop* means synonym and dropouts together.

3.2 VAT Regularization:

Virtual adversarial training extends supervised learning to a semi-supervised setting, such that it can avoid over-fitting to a large extent. The first adversarial attacks data back to 2004 in the context of linear classifiers. In simple words, an adversarial example is a carefully chosen input (via optimization) to fool the system and maximize the prediction error thus getting misclassified. Goodfellow et al.[12] argued their existence to be a consequence of the piece-wise linearity of deep learning systems.

This introduces the majority of machine learning algorithms to new vulnerabilities and security concerns. However, we do not intend to provide a robust model for adversaries. The core idea of this approach is to first generate adversarial examples based on gradients of Kullback-Leibler divergence of the output distribution for similar input distribution and further re-train the model parameters for this input. The intuition of this approach can be seen as having a smoother cost function in a high dimensional setting.

Core Idea

The approach is inspired by the domain of computer vision and faces few limitations in natural language processing due to basic the nature of words being discrete tokens. Thus, finding these adversarial examples with a gradient-based approach is not possible. Inspired from Takeru et al.[11] we opted to introduce perturbations in the embedding layer of the language model and leverage weights from existing pre-trained models like (GloVe[5], word2vec[6], FastText[7], BERT or any custom fine-tuned embedding matrix).

Our approach stays invariant to all different word vector models and converges faster as we do not train the weights in the embedding layer. Next, we find some random disturbance in the vector of a word controlled by a hyper-parameter ϵ to regularize the word representation based on gradient such that it tends to fool the model, and further update the model parameters for a fixed set of epochs. During the last optimization, it is crucial to freeze the parameters that were used to find adversarial input itself, otherwise, the inputs would keep changing and the approach would not make any sense.

Deciding these perturbations, such that is imperceptible to humans in the context of NLP is a difficult problem as it involves semantic relationships in a natural language. The perturbed word vectors might not correspond to any respective word in the embedding model; but that is of our least concern, as the true word vectors themselves face a bigger bottleneck of being context-free (except for recent models like ELMo[8], GPT[9] and BERT[10]). For example:

1. mouse(rodent) and mouse(computer) should ideally have two word vectors. However, traditional models represent them as one which always stops the model from accurate predictions.

[5]https://nlp.stanford.edu/projects/glove/
[6]https://radimrehurek.com/gensim/models/word2vec.html
[7]https://fasttext.cc/
[8]https://arxiv.org/pdf/1802.05365.pdf
[9]https://huggingface.co/transformers/model_doc/gpt.html
[10]https://github.com/google-research/bert

2. A word can have multiple synonyms which is computationally very expensive to query from the embedding matrix and train the deep neural network on all sets of combinations.

VAT does not provide any guarantee of true prediction on all possibilities of rephrasing a single document as we humans can, but it is a regularization methodology with an additional loss term which is calculated irrespective of true labels. Thus it performs well on less amount of training data to predict large test data with a decent accuracy when compared to other supervised approaches. Another advantage could be seen in the aspect of adversarial attacks, as our approach has split the embedding layer from the rest of the model. We can consider it robust to some extent because an adversary usually has no access to the embedding layer under a threat model. However, the provable part of this robustness is a future work.

Working

The model can work as a functional component i.e loosely coupled with its own tokenization or learning word vector representations and expects following inputs for its working,

- clean training set $(\mathcal{X}, \mathcal{Y})$ as discrete token sequences of equal length; it is important for the test set to have similar format.
- Word index $\mathcal{W}$ for all the possible tokens of the vocabulary. The words from a different domain that are not found in $\mathcal{W}$ are given a token value of 'UNK' with zero word vector.
- Custom embedding matrix $\mathcal{M}$ for $\mathcal{W}$ taken as a subset from any existing pre-trained or a fine-tuned model.

The input for the model is a batch of sequences of words represented in three dimensions (batch size, doc length, embedding dim). When piped across *k*-layer network h_θ we can expect an output distribution for any input, also referred as *logits*. Traditional adversarial training considers the true output distribution for an input with these *logits*, further calculates a loss term and performs gradient ascent in order to find a worst case input that would make the model mis-classify itself to some other label. Next, use these adversarial inputs to retrain the network parameters $\boldsymbol{\theta}$ to reduce the total classification loss via gradient descent.

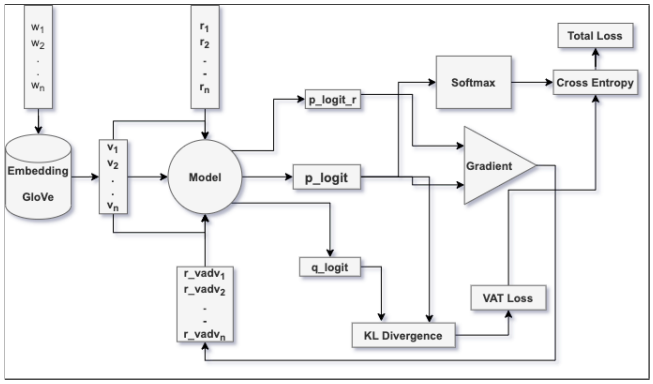

Figure 5: VAT Block Diagram

In the above Figure, **v** is the vector representation of a sequence of words, **r** is a noise controlled by a norm [-ϵ, ϵ]. Both these inputs are parallelly passed across h_θ and a difference in the distribution is calculated through KL divergence metric. Further, this scalar quantity is used to adjust vector **r** such that the loss is maximized. It is crucial to ensure this gradient falls under Δ; all allowable set of perturbations, such that it holds the semantic structure of the original input and we usually bound it in a norm ball like L_2,L_∞ regularization. Later, add this gradient to vector **v** to find KL divergence between *p-logit* and *q-logit*, also termed as our scalar **vat loss** for the whole batch. The main takeaway should be to understand on this *vat-loss* which was never calculated from the true labels but some arbitrary output distribution, and the intention was to reduce the gap of two output distributions for two very similar input distributions. The intuition can be seen as trying to reduce the extreme non-linearity of the neural model in a high dimensional space.

Formally, let us denote $\boldsymbol{v}$ as our original input, which represents a sequence of word vectors $[v^1, v^2, ..., v^T]$; where T is the time-step or sequence length. We propose intruding a small random vector $\boldsymbol{r}$ in the embedding space with same dimensions as the input embedding. The classifier h_θ has the weight parameters $\boldsymbol{\theta}$, and $\hat{\boldsymbol{\theta}}$ for finding finding adversarial inputs i.e. a copy of $\hat{\boldsymbol{\theta}}$ ensures that during backpropagation the perturbations should not change. Next, we define our language model conditional probability for the label $\boldsymbol{y}$ given $\boldsymbol{v}$.

$$p(. \mid \boldsymbol{v}; \hat{\boldsymbol{\theta}}) \tag{4}$$

$$\boldsymbol{g} = \nabla_{\boldsymbol{v}+\boldsymbol{r}} \, \boldsymbol{KL}[p(. \mid \boldsymbol{v}; \hat{\boldsymbol{\theta}}) \,||\, p(. \mid \boldsymbol{v} + \boldsymbol{r};\, \hat{\boldsymbol{\theta}}] \tag{5}$$

Equation 5, calculates the gradient of two output distribution where r is a TD dimensional vector with T - *sequence length* and D - *embedding dimension*. Next, we formulate the following $r_{vadv} = \epsilon \mathbf{g}/||\mathbf{g}||_2$ to calculate our adversarial input.

$$\boldsymbol{L_{vadv}(\theta) = KL}[p(.|\boldsymbol{v}; \hat{\boldsymbol{\theta}})||p(.|\boldsymbol{v} + \boldsymbol{r_{vadv}}, \boldsymbol{i}; \boldsymbol{\theta}] \tag{6}$$

The virtual adversarial loss, $L_{vadv}(\theta)$ can be considered for both labelled and unlabelled data in the training set. This loss forces the model to bring same output distribution for the adversarial input as it got for the original input.

Experimental Setting

The model architecture comprises of an independent Embedding layer, a BiLSTM as the first hidden layer with 128 units, followed by two dense layers with 64 and 32 *tanh* activation units respectively. This network architecture outputs logits for all three sets of inputs required during model training. Finally, the model is build on a output layer comprising of a dense with 2 activation units which performs a softmax operation on these logits. In case of multiple classes, one can just change the number of units in the output layer.

The optimizer used is Adam and categorical cross entropy loss along with the scalar vat loss contribute towards the net loss. The model is dependent on the quality of data to a large

Algorithm 2: Virtual Adversarial Training

Data: train set $(\mathcal{X}, \mathcal{Y})$, learning rate η, noise norm ϵ, network h_θ, embedding matrix $\mathcal{M}$, word index $\mathcal{W}$
Result: Classify each document as fake or legit.
initialization; training batch β, parameters θ
for *x,y in* $(\mathcal{X}, \mathcal{Y})$ **do**
1. Calculate embedding; $\mathcal{X}_\mathcal{E}$ = $\mathcal{M}$(**x**)
2. Get first logits for true input; *p-logit* = $h_\theta(\mathcal{X}_\mathcal{E})$
3. Add perturbations; $\mathcal{R}_\mathcal{E}$ = $\mathcal{R}_\mathcal{E}$ + $\mathcal{X}_\mathcal{E}$; $|\mathcal{R}_\mathcal{E}| \leq \epsilon$
4. Second logits for noised input; *p-logit-r* = $h_\theta(\mathcal{R}_\mathcal{E})$
5. Gradient: g = $\nabla_{\mathcal{R}_\mathcal{E}} KL[$*p-logit, p-logit-r*$]$;
6. Adversarial Input; $\mathcal{V}_\mathcal{E}$ = $\epsilon \mathbf{g}/||\mathbf{g}||_2$
7. Final logits; *q-logit* = $h_\theta(\mathcal{X}_\mathcal{E} + \mathcal{V}_\mathcal{E})$
8. $L_{vadv}(\theta) = KL[$*p-logit, q-logit*$]$;
for $n \leq n_{epochs}$ **do**
$\theta := \theta - \frac{\eta}{|\beta|} \sum_{i \in \beta} \nabla_\theta L(h_\theta(x_i), y_i) + L_{vadv}(\theta)$
end
end

extent; for example the incorrect words like spelling errors, clustered words if not removed would confuse the model from right predictions. Next, the model was tested on two sets of pre-processing.

- Same pre-processing as other approaches in this paper, where the vocabulary size is 27664 and 11820 (43%) words could not be found due to stemming, and were all initialised with zero as their vector representation.
- Pre-processing without stemming and only converting word in its morphological form or its lemma. Vocabulary size 38028 and 6210 words not found in GloVe (16%) data loss. The model observed $\tilde{3}$% improvement in the test accuracy.

3.3 Pseudo Label

Pseudo labels are the labels for unlabelled data which are treated as a real labels, these labels acts as the classes with the maximum predicted probability.

For our binary classification it will be assigned using tri-training approach suggested in paper [10]. Tri-training is originally used for domain adaption where a common architecture is shared between multiple domains and the two predictors will be train for predicting pseudo labels and third classifier will be trained only using pseudo labels to get domain specific representation. Tri-training was further applied in detailed on sentiment classification task by Ruishan Liu , Liyue Shen. [13]. In our approach, we have made a change in original approach of tri-training, instead of training the third model with the all new updated data set, we have used the already trained predictors to evaluate on validation set as we are only focusing on fake news classification i.e. Single domain. We evaluate the validation set on both of the predictor model and taken the result from the model which gives better performance among two. Thus, making it as a bi-training for pseudo label inspired from tri-training[10].

Algorithm 3: Bi-Training for Pseudo Label

Data: train set $(\mathcal{X}, \mathcal{Y})$
while *Epochs* **do**
1: Train two predictor models simultaneously only using labelled data;
$\mathcal{M}_\mathcal{A}, \mathcal{M}_\mathcal{B}$
2: Use this supervised model to predict pseudo labels for unlabeled data;
3: Select the labels for unlabelled data based on two conditions:
I- Both Predictors should predict same label
II- Confidence score for both predictors should exceed the threshold $(\mathcal{T})$
4: Add the Selected Samples and Pseudo Labels to training set and retrain both the predictors ;
$\mathcal{X} = \mathcal{X} \cup \mathcal{X}_{Select}$
$\mathcal{Y} = \mathcal{Y} \cup \mathcal{Y}_{Pseudo}$
5: Evaluate on validation set. $(\mathcal{M}(\mathcal{X}_\mathcal{V}, \mathcal{Y}_\mathcal{V}))$
end

Deciding confidence threshold $(\mathcal{T})$
Confidence threshold is the core part of pseudo labeling, it is % of confidence that our predictors are confident about selected label. As observed that if we don't define it, many samples can be classified wrongly just because their probability will be a bit higher than other labels. We can see a case where for example: predicted probabilities are 0.4999 for Label 0 and 0.5001 for Label 1. In such cases it's hard to decide which is true class for given sample and our classifier may predict wrong label.

As observed in paper for sentiment classification using Pseudo-Labels [13], we can notice that increasing the confidence threshold $(\mathcal{T})$=0.99 can lead to lower number of pseudo-labels and high quality of pseudo labels. On other hand when we decrease the threshold $(\mathcal{T})$= 0.6 the quantity of pseudo labels are increasing but the quality of those labels are decreasing. In sentiment classification task, we can see the trade off between quantity and quality of Pseudo Labels is stable around threshold $(\mathcal{T})$ between 0.85 and 0.90. We have focused on quality side and planned to keep threshold as $(\mathcal{T})$ = 0.90. our model starts learning pseudo labels after 5 epochs when the threshold is set to 0.90.

When the threshold is set to high value for ex. 0.9 or 0.99, model predicts less number of pseudo labels so it will take more number of epochs to get pseudo labels for all unlabelled data. On opposite side, if threshold is low for ex. 0.6, model will predict pseudo labels in less than 10 epoch for every unlabelled data but as always the quality of those pseudo labels will be very low compare to the one with high threshold value.

Predictor model architecture $\mathcal{M}_\mathcal{A}, \mathcal{M}_\mathcal{B}$
Both the predictors shares same architecture, the architecture includes one embedding layer, one BiLSTM layer with 128

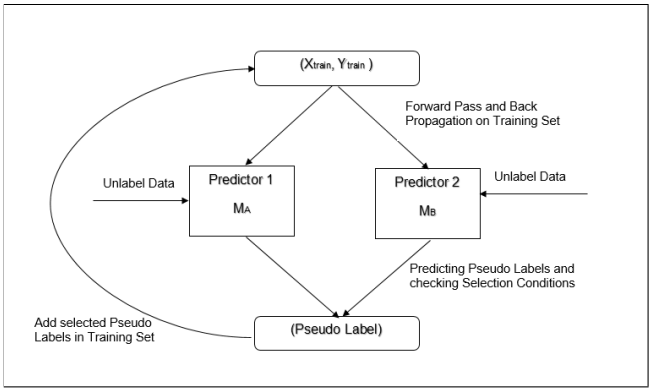

Figure 6: Bi-training Pseudo Label Network Structure

hidden units followed by dense layer with 2 units and the output layer with softmax activation and 2 unit. As we want to check the confidence threshold for pseudo labels and categorical cross-entropy used as loss for both the predictor. Finally the original labels has been one hot encoded to work with loss function. Figure 6 shows bi-training structure for training and predicting pseudo labels.

Noise approach

There were no noise approach introduced in paper [13] and it was the point for further improvement to avoid the over-fitting of predictors. We have taken the approach of adding noise as specified in exploration of noise strategies in semi-supervised named entity classification[14]. One of the noise approach which they have used in Semi-supervised Mean-Teacher model was Word Dropout. They are dropping K number of word tokens to get best from the model and prevent the over-fitting on training set. We have taken this noise generation approach for classifying fake news in our bi-training Pseudo Label model before embedding, where we are randomly dropping the tokens by replacing it with zero to drop them. we have set number of words to drop (K) as 5 which was already performing best [14]. We only added this noise in original training set and not in the pseudo labeled samples which further selected and added in training set of predictors. There might be still the case that model will be over fitted and learns completely about noisy data.

4 Experimental Data

We have selected the benchmark datasets which has articles of different timelines (2016-2019), and we have mainly focused on the data from political aspect for our problem domain. Dataset 2 table is given below and others datasets in appendix.

- **NELA-GT-2019:** It contains 1.2 M news articles from 1-Jan-2019 to 31-Dec-2019. This dataset includes the ground level truth labels from 7 different news veracity assessment sites. We found out that our models were over fitting with these data, hence we kept these data as unlabel data. The summary of dataset is mentioned in Dataset 1 table [15].
- **LIAR:** It includes 12.8K human labeled short statements from politifact.com. In our work, we have tried to differentiate real news from all types of hoax, propaganda, satire and misleading news. Hence, we have focused on classification of news as real and fake. For the binary classification of news, we have kept only the true and false labeled news. This dataset mostly deals with data from political domain that include statements of democrats and republicans, as well as a significant amount of posts from online social media [16].
- **Credibility:** It is also a publicly available dataset that has been used. They have used the two public datasets for fake news detection from Buzzfeed News and PolitiFact. Besides news content and news labels (i.e., fake, or true), the datasets contain information on the social networks of users involved in spreading the news [17].
- **FakeNewsNet:** It has multi-dimensional information related to news content, social context, and spatiotemporal information. The news content contains PolitiFact and GossipCop datasets [18].
- **BERT:** The data used in BERT contains 904 articles and we considered these articles as unlabeled data [19].
- **Snopes & Kaggle:** Snope dataset consists of rumors analyzed on along with their credibility labels (true or false), sets of reporting articles, and their respective web sources. Kaggle Article dataset contains the New York Times, Breitbart, CNN, Business Insider, the Atlantic, Fox News, Talking Points Memo, Buzzfeed News, National Review, New York Post, the Guardian, NPR, Reuters, Vox, and the Washington Post articles.

Source	True label count	Fake label count	Total	Unlabeled count
Paper-Credibility	209	211	420	-
Site-Political	752	841	1593	-
Paper-NELA-GT-2019	-	-	-	2365
Paper-Bert	-	-	-	904
Total	**963**	**1050**	**2015**	**3269**

Table 2: Dataset 2

Overview of the pre-processing steps which are applied on the dataset is shown in Figure 7. Pre-processing is followed by the conversion of Text data into vectors and pad sequence generation.

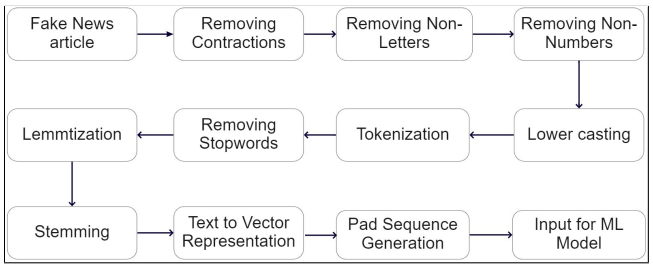

Figure 7: Overview of Data Pre-processing

Data Exploration

Out of the mentioned datasets, we found out that Dataset 2 was performing well as compare to other datasets without the problem of over-fitting and under-fitting. Below are the data exploration analysis for datasets 2 and rest of data exploration analysis are given in the Git hub location and few are mentioned in Appendix [11].

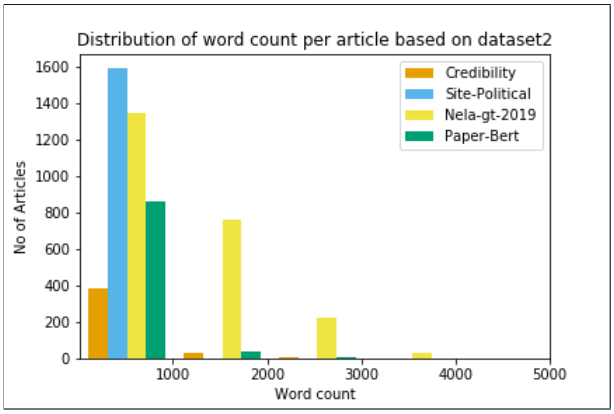

Figure 8: Distribution of word count of dataset 2

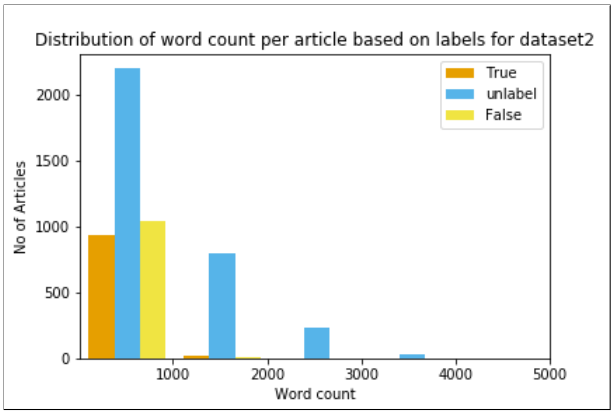

Figure 9: Distribution of word count per article based on labels of dataset 2

5 Evaluation

In this section we have individual observation of the proposed approaches and then in the end overall comparison of the proposed approach is discussed.

5.1 Mean Teacher

To compare the result of the student model with the teacher model in different epoch shown in the table 3. In table 3, we are just comparing the student and teacher model performance in one of the mentioned strategy i.e synonym replacement. Model ran on the experiment setting mentioned in table 1 in google colab. We used dataset 2 for the experiment and size is mentioned in table 9. As per observation, after particular epochs, the teacher model start performing better than the student model in this case at epoch 15, where teacher model starts overtaking the student model. The convergence of teacher model depends on epoch, batch size, train data size, and alpha α. Tuning of parameter is required to get better result.

The impact of different noise strategy on performance of Mean teacher model is shown in table 6. Comparatively, synonym replacement and drop out have out performed other noise strategies including supervised BiLSTM model and label propagation. And, same is reflected in ROC curve 10. Synonym replacement has outperformed dropout strategy if we consider precision in detecting true news i.e. 0.700 ± 0.03 and 0.691 ± 0.03 respectively. And, binary loss for synonym is 0.552 ± 0.03 which is comparatively lower than dropout 0.557 ± 0.02. However, overall dropout have better recall and f1 score in detecting true news class as well as better precision for fake class. As per ROC curve, synonym curve is more aligned towards upper left corner as compared to dropout and other strategies.

As of now, the probability used in both the strategy we used single value as shown in table 1 but there may be chances at different probabilities both model can show much better result or dropout might perform better than synonym replacement. Surprisingly, combination of both synonym and dropout has not shown result we were expecting before experiment and performed worst. There may we need to tune the probability to get the expected result from this approach. There is future scope for us to measure performance at different probabilities value and find out best performing value. There is another observation related to alpha α, lower the value teacher model converges faster and vice versa. Mostly, configure alpha α 0.99 but once the teacher model converges then ramp down the value to 0.999 to achieve better result. Implementing ramp down of alpha is still a scope of future work. Additionally, worth noticing observation of this experiment is dropout strategy is comparatively faster or less time consuming than any other approaches.

Comparison with VAT and Pseudo label is discussed in 5.4.

5.2 Pseudo Labelling

Evaluation on Pseudo Label has been done by considering the amount of labelled and unlabelled data. The number of labelled data considered was **200** samples with equal class proportion and the amount of unlabelled data used was around **2000** samples. For validation we have taken around 600 samples. We have trained the supervised BiLSTM model which has same architecture as both predictors. The supervised BiLSTM was trained with only 200 labelled data and the Bi-Training pseudo label was trained using both labelled and pseudo labelled data. The results in table 4 shows the performance of both model on validation set and Fig 11 shows the ROC curve for the pseudo label. We can notice

[11] https://github.com/bksaini078/fake_news_detection

Model	Test Accuracy	Precision	Recall	F1-score	Binary loss
Epoch 10					
Student	0.703	0.697	0.765	0.725	0.572
Teacher	0.657	0.677	0.728	0.660	0.664
Epoch 15					
Student	0.695	0.710	0.701	0.703	0.747
Teacher	0.702	0.700	0.750	0.721	0.552
Epoch 20					
Student	0.706	0.727	0.696	0.710	0.812
Teacher	0.704	0.707	0.736	0.720	0.649
Epoch 25					
Student	0.697	0.703	0.716	0.708	0.994
Teacher	0.703	0.706	0.726	0.715	0.807

Table 3: Performance of student and teacher model at different epochs(synonym replacement). Showing average value of each parameter for predicting true news in 10Kfold cross validation.

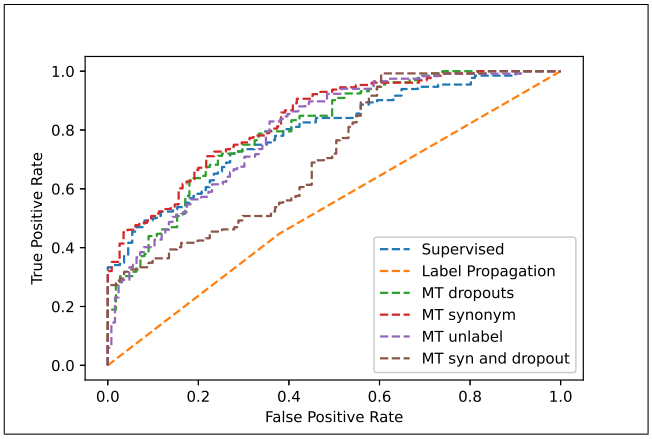

Figure 10: ROC curve for supervised, label propagation and Mean teacher with different noise strategy.

that with minimal **labelled data around 10% and 90% unlabelled data** on average of 10 iteration our Bi-Training pseudo label performs better then supervised model. The results are the mean result of 10 iteration on validation set with $\pm Confidence\_interval$ of result with 95% confidence interval.

Model settings: batch size-64, epoch-14, Confidence Threshold ($\mathcal{T}$) - 0.90 , Labelled data- 200, Unlabelled data-2000.

Table 6 shows result on 10 KFold validation, with all labelled data and 600 unlabelled data.

5.3 VAT regularization

To be able to better compare all the approaches on a common benchmark the reports are presented in the evaluation table [5.4] are for common set of vocab size and pre-processsing. The first table below provide the test results over a very small training size to predict large test data in similar domain. These evaluations are promising for future work and more experiments need to be performed on unlabelled datasets or where data annotation is very expensive. VAT seems to require less training data to give equal results as other supervised approaches.

Matrices	Supervised	Pseudo Label
Test-Accuracy	0.550 $\pm$0.02	**0.579 $\pm$0.03**
Precision-True	0.632 $\pm$0.02	0.663 $\pm$0.02
Precision-Fake	0.508 $\pm$0.02	0.541 $\pm$0.03
Recall-True	0.426 $\pm$0.08	0.491 $\pm$0.13
Recall-Fake	0.698 $\pm$0.07	0.685 $\pm$0.10
F1-True	0.496 $\pm$0.06	0.534 $\pm$0.10
F1-Fake	0.583 $\pm$0.02	0.592 $\pm$0.02
loss	1.420 $\pm$0.24	1.365 $\pm$0.42

Table 4: Performance of Pseudo Labels on validation set with 200 labelled data and 2000 unlabelled data.

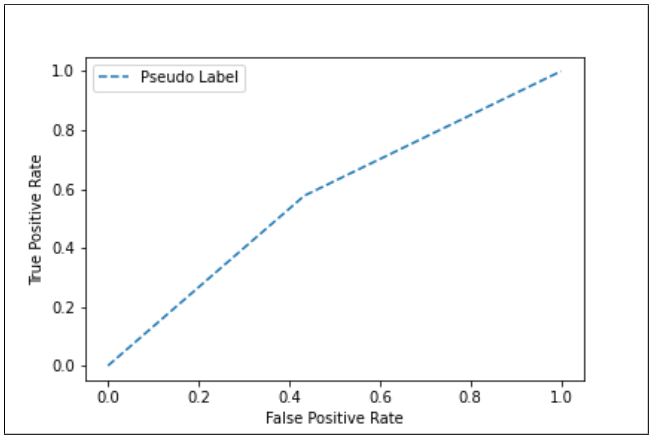

Figure 11: ROC curve for Bi-training Pseudo Label model with 200 labelled data and 2000 unlabelled data

ϵ	k-fold	train size	test size	mean test accuracy
0.02	2	1217	1216	0.670
0.02	**3**	**811**	**1622**	**0.656**
0.01	3	811	1622	0.641
0.01	4	486	1947	0.625

The results in the table 5 are on word tokens without the stemming operation which gave a better representation of the training data as the data loss decreased from 40% to 16% unknown words for model learning. The later occurs mainly due to the reason of noisy words, spelling errors, etc. The experimental setup as mentioned in the above table stays uniform for all the VAT results in this paper. However, on lowering the batch size to 32 an increase in the performance was observed. Another, conclusion for VAT is a faster convergence, so in order to avoid over-fitting the model was restricted to 8 epochs unlike other approaches in this paper.

5.4 Overall Comparison

In detecting fake news, we experimented with all the proposed techniques with same setting mentioned in table 1 and we evaluated all the model observed that Bi-training pseudo label, mean teacher with synonym and VAT techniques performed better than other supervised or weakly supervised proposed techniques. Pseudo label approach has shown better

Metric	Score
Test Accuracy	0.731 ± 0.02
Precision	0.734 ± 0.04
Recall	0.730 ± 0.03
F1 Score	0.732 ± 0.04
Binary Loss	0.529 ± 0.08

Table 5: lr = 0.2e-3, batch size = 64, epochs = 8, C.I = 0.95

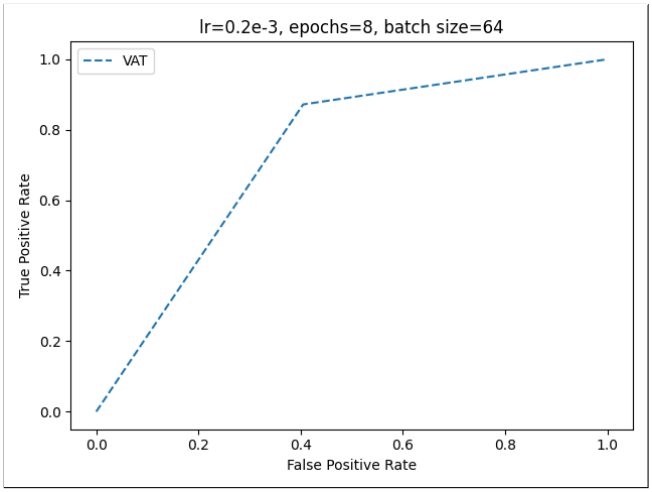

Figure 12: ROC curve for Virtual Adversarial Training(16% loss)

test accuracy i.e. 0.731 ± 0.01, better precision in detecting true and fake news i.e. 0.718 ± 0.03 and 0.756 ± 0.02 respectively, recall for true i.e. 0.797 ± 0.04,and F1 score for true and fake i.e. 0.753 ± 0.02 and 0.698 ± 0.03 respectively. Mean teacher with synonym technique has lowest binary loss i.e. 0.552 ± 0.03 followed by mean teacher with synonym. Whereas, VAT model has shown better result in recall true 0.662 ± 0.07. As per ROC (Receiver operator characteristics) 13, pseudo label model is more lien towards upper left corner and comparatively performed well in detecting with better true positive rate and lower false positive rate. However, as in VAT there is 40% loss of vocabulary but as observation if loss is negligible then VAT out perform every other strategies in terms of test accuracy and precision, glimpse of the result with 16% loss can be seen in table 5. Overall, in fake news detection with weakly supervised learning pseudo label, mean teacher with synonym and VAT with less loss can perform better. One of the reason for pseudo label outperforming other proposed model can be efficient utilization of unlabelled data as compared to other proposed model. As, mean teacher model only utilizing unlabeled data in vocabulary and distribution difference however, mean teacher with dropout which significantly performed better is only a supervised approach. And, mean teacher with synonym noise demands significant amount of unlabeled data to increase the vocabulary hence syntactic accuracy. On the other hand, pseudo label is treating it as input for training

6 Conclusion

In our attempts to implement weakly supervised learning for fake news detection, we have experimented on multiple combinations of existing approaches and a few novelties. Considering the small training size, we believe our results are very promising for pseudo labels and mean teacher. Pseudo label performs good with less labelled data. In a nutshell, all these methodologies do not bet completely on the true labels for the model training, which makes it an ideal choice for fake news classification. We can draw two main advantages from this, less over-fitting and overcoming the issue of expensive data annotation where supervised learning approaches are saturating.

Next, the free media platforms does not promise data validation and, texts or documents consists of many noisy words which were presently used for our model training. This has been a major confusion for the models, as they were never replaced with their correct representation. However, in an ideal setting it is not expected to have a high data quality and so the right word vector representation is very crucial. Garbage in, garbage out.

Our test results, without replacing these hammy words for the pseudo label saturates at 0.73 test accuracy. The several regularization methodologies discussed above in mean teacher model has given the best scores. However, our conclusion stays with Pseudo label and VAT which has certainly out performed all other approaches seeing its test results in a high constrained setting [Table 5]. The main takeaway for both, is it requires very few training data to give best relative results, faster convergence, and less hyper-parameter tuning.

6.1 Future work

This line of work has many possibilities for future research. We have experimented with multiple existing embedding models, trained them during model development, applied several regularization's (drop-out, synonyms), and implemented gradient based adversarial inputs with a goal of implementing semi-supervised learning. We would like to list the following as a continuation to our work:

- De-noising the vocabularies in the corpus, i.e use predictive models to replace error words (spelling mistakes, missing spaces between tokens, acronyms, etc.)
- Experiment these approaches on larger training datasets, as we believe the test results are bound to increase. In proposed approaches, we have utilized only syntactic characteristics of Natural languages and in future, there is scope for introducing semantic based noise which is primarily created for natural language processing domain to achieve better result. During research,we found out many different combination like dropout and Unlabeled distribution together as classification and consistency cost, teacher model first learn from student and regularizes with VAT, only Mean teacher with continuous ramping down of the alpha, exponential moving average weights assignment should only happen when student model perform better than last prediction, and many more. And, would like focus on combining the strategies and observing the result.

Model	Test-Accuracy	Precision-True	Precision-Fake	Recall-True	Recall-Fake	F1-True	F1-Fake	Binary loss
Supervised-BiLSTM	0.687 ±0.02	0.694 ±0.02	0.692 ±0.03	0.728 ±0.05	0.648 ±0.07	0.706 ±0.02	0.663 ±0.03	0.773 ±0.09
Label Propagation	0.505 ±0.02	0.528 ±0.03	0.490 ±0.03	0.397 ±0.04	0.622 ±0.02	0.452 ±0.03	0.548 ±0.02	7.77 ±0.06
Pseudo Label	**0.731 ±0.01**	**0.718 ±0.03**	**0.756 ±0.02**	**0.797 ±0.04**	0.656 ±0.06	**0.753 ±0.02**	**0.698 ±0.03**	0.721 ±0.08
MT Synonyms	0.702 ±0.02	0.700 ±0.03	0.710 ±0.03	0.750 ±0.04	0.649 ±0.06	0.721 ±0.02	0.674 ±0.03	**0.552 ±0.03**
MT Unlabeled	0.629 ±0.05	0.629 ±0.03	0.689 ±0.13	0.695 ±0.22	0.561 ±0.17	0.641 ±0.10	0.585 ±0.07	0.584 ±0.02
MT Dropouts	0.702 ±0.01	0.691 ±0.03	0.751 ±0.06	0.786 ±0.08	0.607 ±0.10	0.729±0.02	0.653±0.05	0.557±0.02
MT Syn and Dropouts	0.636 ±0.04	0.684 ±0.07	0.674 ±0.09	0.660 ±0.19	0.612 ±0.17	0.623 ±0.11	0.602 ±0.07	0.598 ±0.05
VAT (40% token loss)	0.701 ±0.03	0.703±0.05	0.702 ±0.04	0.737 ±0.05	**0.662 ±0.07**	0.718 ±0.68	0.678 ±0.04	0.598 ±0.07

Table 6: Evaluations of the approaches . *lr* = 0.1e-3, batch size = 64, C.I = 0.95

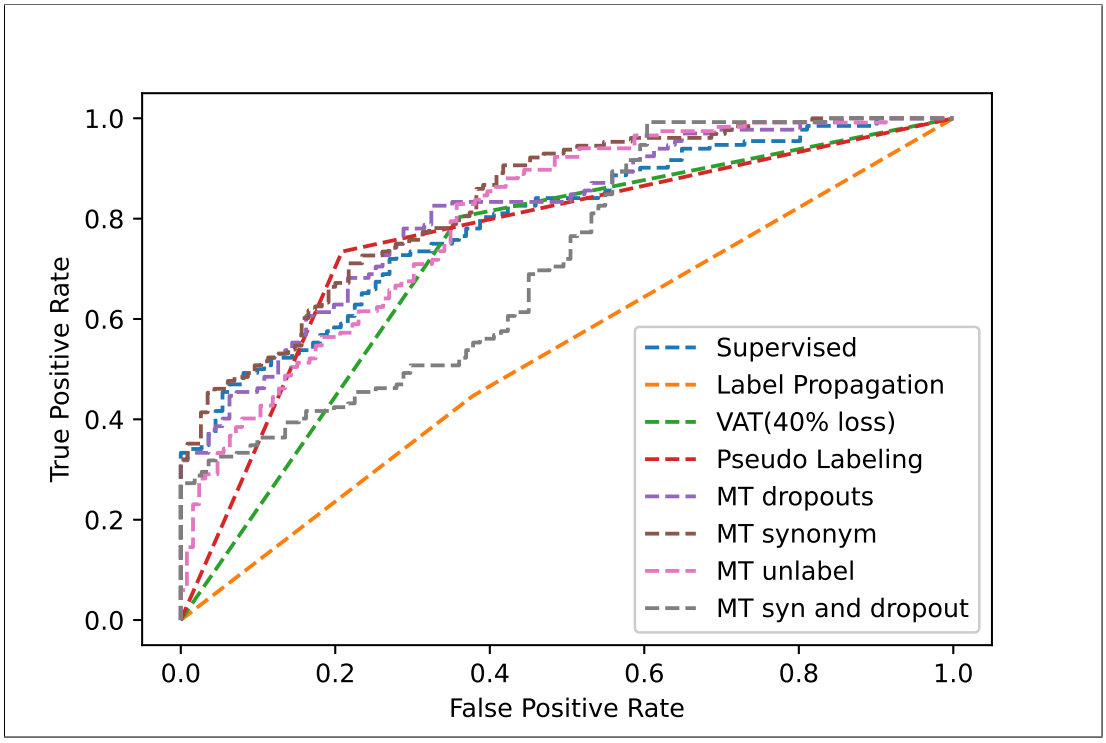

Figure 13: ROC curve for supervised, label propagation, Mean teacher with different noise strategy, Psuedo labelling, and VAT

- Usage of recent embedding models, which are context dependent word vectors. This allows a better representation of natural language and would clear the biggest bottleneck we have faced in our experiments.
- An attempt to bridge the gap between mean teacher regularization methods (synonyms) and VAT, by introducing perturbations in a similar direction rather than a random vector controlled by a continuous variable.
- We believe a promising direction could be an attempt to leverage attention models to introduce perturbation for VAT. If combined with context dependent vectors, this would make the language models behave very close to how humans understand natural languages.
- In Bi-training pseudo label model we have observed few cases where throughout complete training, predictors doesn't predict any pseudo labels at all when the threshold is high making it as a supervised model. For this, one simple solution we thought of is to just increase the number of epochs until model doesn't learn pseudo labels, we have to think some efficient way to overcome this. Also, we have only added word dropout noise to original training set and not on pseudo labeled set, we have observed few cases of over fitting on noisy train set, we are planning to apply some further noise approach to pseudo labelled data.

References

[1] Axel Gelfert. Fake news: A definition. *Informal Logic*, 2018.

[2] Melanie Freeze, Mary Baumgartner, Peter Bruno, Jacob Gunderson, Joshua Olin, Morgan Ross, and Justine Szafran. Fake claims of fake news: Political misinformation, warnings, and the tainted truth effect. *Political Behavior*, 02 2020.

[3] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 08 2017.

[4] Kai Shu, Suhang Wang, Dongwon Lee, and Huan Liu. Mining disinformation and fake news: Concepts, methods, and recent advancements. *Disinformation, Misinformation, and Fake News in Social Media*, page 1–19, 2020.

[5] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, 2013.

[6] Nitish Srivastava. Improving neural networks with dropout. *University of Toronto*, 182(566):7, 2013.

[7] Yaqing Wang, Weifeng Yang, Fenglong Ma, Jin Xu, Bin Zhong, Qiang Deng, and Jing Gao. Weak supervision for fake news detection via reinforcement learning. *arXiv preprint arXiv:1912.12520*, 2019.

[8] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. pages 797–806, 11 2017.

[9] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[10] Tatsuya Harada Kuniaki Saito, Yoshitaka Ushiku. Asymmetric tri-training for unsupervised domain adaptation. *arXiv:1702.08400*, 2017.

[11] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification, 2016.

[12] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014.

[13] Liyue Shen Ruishan Liu. Unsupervised domain adaptation for sentiment classification using pseudo-labels.

[14] Pooja Lakshmi Narayan, Ajay Nagesh, and Mihai Surdeanu. Exploration of noise strategies in semi-supervised named entity classification. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 186–191, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[15] Jeppe Norregaard, Benjamin D. Horne, and Sibel Adali. NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles, 2019.

[16] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[17] Niraj Sitaula, Chilukuri K. Mohan, Jennifer Grygiel, Xinyi Zhou, and Reza Zafarani. Credibility-based fake news detection. *ArXiv*, abs/1911.00643, 2019.

[18] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *CoRR*, abs/1809.01286, 2018.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

7 Appendix

- **Ravineesh Goud:** Abstract, Introduction, Background and Related Work, Mean teacher and VAT technical details, Research paper collection for references, bibtex creation, Proposal presentation, Proposed Approach, Experimental Data, Paper outline creation, Review.
- **Swathi Chidipothu Hare:** Introduction, Background knowledge,Proposed Approach, Experimental Datasets, Collection of datasets, Prepossessing of datasets, Exploration analysis, Code-Python packages and Parameterized for all models (MeanTeacher, VAT, Pseduo Label), Data pre-processing(code),supervised model(code), Integration of code with all models and tested with different datasets,Github repository,Paper outline creation and Review
- **Bhupender Kumar Saini:** Methodology, Background and Related Work, Proposed approach, Dataset collection and testing at the initial stage , Mean teacher Algorithm with different noise strategies(Research, initial prototype development, final code, diagrams, and technical details), VAT (initial prototype development and research), Label propagation(code and training model), Evaluation, Report creation, Overall comparison(including table and ROC curve), Data preprocessing(Code), Github repository, Conclusion, Future work, and review.
- **Mayur Waghela:** Methodology, Related work,Proposed approach, Mean teachder with unlabelled data(Research, Report, Algorithm and code), Data pre-processing(code),supervised model(code), Pseudo Label(Research on Bi-training approach, Algorithm, code, report, diagrams), Evaluation, conclusion, Future work.
- **Lokesh Sharma:** Methodology, Virtual Adversarial Training, data preprocessing in spacy, Evaluation, Test Results, Conclusion & Future work

7.1 Other Dataset tables and Data Exploration Analysis :

Dataset 1	True label count	Fake label count	Unlabeled count
Politicususa	-	-	4048
skynewspolitics	-	2269	-
Buzzfeed	1741	-	-
Politico	2388	-	-
Total	**4129**	**2269**	**4048**

Table 7: Dataset 1.

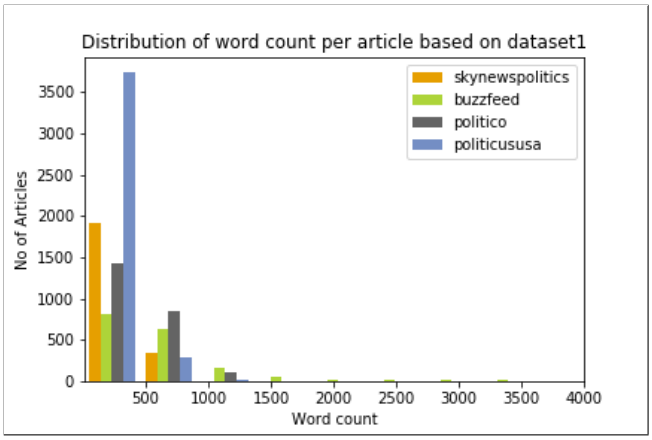

Figure 14: Distribution of word count of dataset 1

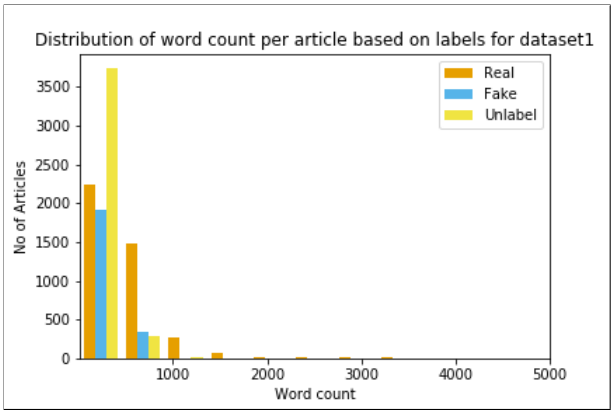

Figure 15: Distribution of word count per article based on labels of dataset 1

Source	True label count	Fake label count	Total	Unlabeled count
Paper-Credibility	209	211	420	-
Paper-FakeNewsNet	490	490	980	-
Site-Kaggle Article	-	-	-	2500
Total	**701**	**701**	**1400**	**2500**

Table 8: Dataset 3

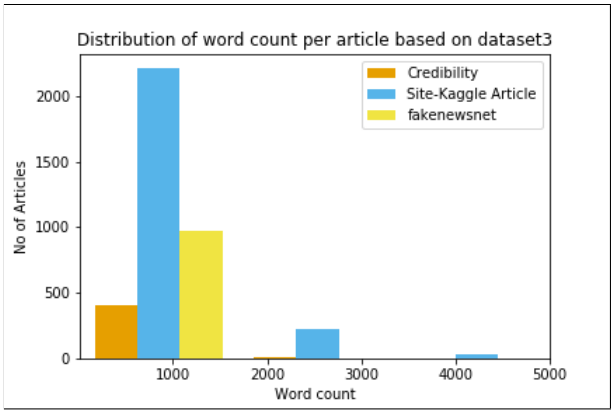

Figure 16: Distribution of word count of dataset 3

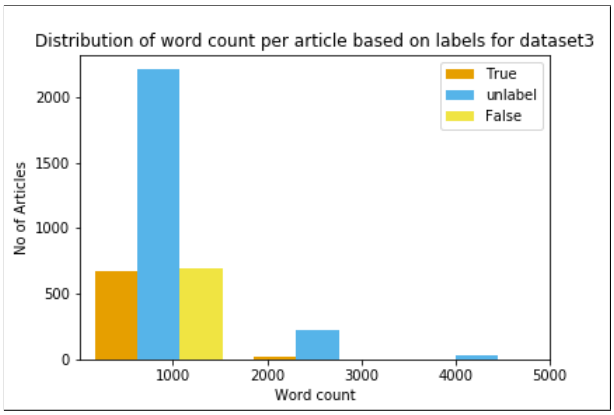

Figure 17: Distribution of word count per article based on labels of dataset 3

Source	True label count	Fake label count	Total	Source	Unlabeled count
Paper-Credibility	197	196	393	Paper-NELA-GT$_2$019	4281
Paper-FakeNewsNet	490	490	980	Paper-BERT	896
Paper-Liar	1670	1979	3649	-	-
Site-Political Data	752	841	1593	Site-Kaggle Article	2996
Site-snops	77	312	389	Site-Kaggle Fake News	897
Total	**3186**	**3818**	**7004**		**9070**

Table 9: Dataset 4

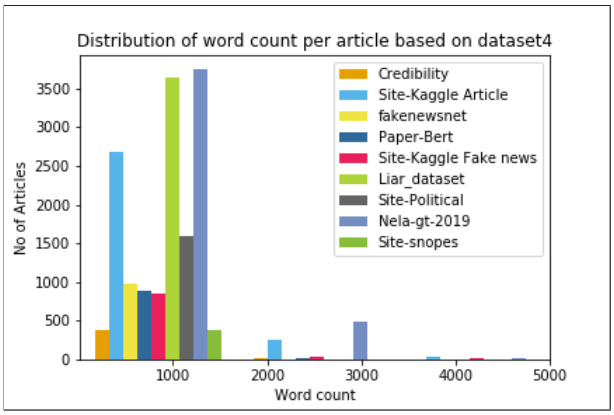

Figure 18: Distribution of word count of dataset 4