Assessment of Mean Teacher and Prominent Adversarial Unlabeled Data for Language Classification

Masterarbeit

Master of Science (M.Sc.) im Web and Data Science

vorgelegt von

Bhupender Kumar Saini

[219 100 887]

Koblenz, im May 2021

Erstgutachter:	Prof. Dr. Andreas Mauthe (Institut für Wirtschafts- und Verwaltungsinformatik, FG Mauthe)
Zweitgutachter:	Alexander Rosenbaum, M. Sc. (Institut für Wirtschafts- und Verwaltungsinformatik, FG Mauthe)

Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden. ja □ nein □

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja □ nein □

....

(Ort, Datum) (Unterschrift)

Zusammenfassung

Die Zusammenfassung Ihrer Thesis.

Abstract

The Abstract of your thesis.

Contents

List of Figures

List of Tables

1 Introduction (1-3 pages)

The machine learning model has proven their advantages in dealing human-specific task and has been widely adopted in the every domain such as autonomous driving, healthcare, banking, manufacturing, logistics, and many more. Furthermore, machine learning model has out performed human capabilities in performing tasks like chess, alpha Go, prediction trends and many more resulted in trend of increasing popularity and dependencies on machine learning model. Especially, Deep Neural Network(DNN) has been widely adopted in real world applications and study in every domain and research field. DNN got upper hand in contrast to any other machine learning algorithm because of its ease of computation at large scale and capability of solving complex problem either linear or non-linear problem [**?**]. To be specific in natural language processing tasks, DNN models also has shown significant advancement in solving various tasks such as text to speech, fake news detection, reviews comment classification and so on. Furthermore, recent works [**?**, **?**, **?**, **?**] in NLP has led to state-of-the-art language model based on BERT, which has been successful across a wide variety of NLP tasks and is consequently the most widely adopted language model. However, several studies have found weaknesses in DNN against adversarial attacks [**?**, **?**, **?**, **?**, **?**], which has drawn substantial attention from the research community.
Introducing a small perturbations in the input image can fool state-of-art deep neural network with high probability and these misclassified samples were called as *Adversarial Samples* [**?**]. Sabotaging machine learning model using adversarial examples are called as *Adversarial attacks* as shown in figure 1.1. These research exposed alarm about the weakness of DNN system against attacks which can raises concern related to user privacy, safety, and security and finally, 'Can we trust ML models?'.
Unfortunately, it is more extensively studied in the domain of computer vision than that of natural language processing [**?**]. And, the implementation of adversarial attacks in NLP presents more challenges due to its discrete nature and the need to maintain semantics [**?**].
And, BERT performance also degrade under adversarial attacks [**?**, **?**]. BERT model performance has shown accuracy lower than 10% under adversarial attack. A less sophisticated spelling error can lead to bad performing machine learning model [**?**] which raises concern on robustness of BERT model.

TODO: More stats on attacks *In real life, people are increasingly inclined to search for related comments before shopping, eating or watching film and the corresponding items with recommendation score will be given at the same time. The higher the score is, the more*

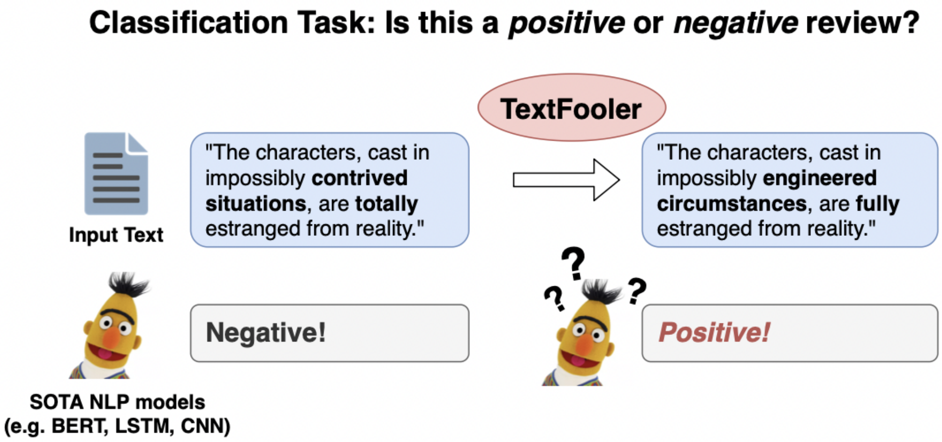

Classification Task: Is this a *positive* or *negative* review?
TextFooler
Input Text
"The characters, cast in impossibly **contrived situations**, are **totally** estranged from reality."
"The characters, cast in impossibly **engineered circumstances**, are **fully** estranged from reality."
Negative!
Positive!
SOTA NLP models
(e.g. BERT, LSTM, CNN)

Figure 1.1: Adversarial attack presented by TextFooler [?], small change in input text influenced the prediction.

likely it is to be accepted by humans. These recommendation apps mainly take advantage of sentiment analysis with others previous comments [20]. Thus attackers could generate adversarial examples based on natural comments to smear competitors (see Fig.1 for instance) or do malicious recommendations for shoddy goods with the purpose of profit or other malicious intents. Apart from mentioned above, adversarial examples can also poison network environment and hinder detection of malicious information [21], [22], [23]. Hence, it is significant to know how adversarial attacks conduct and what measures can defend against them to make DNNs more robust."A survey on Adversarial Attacks and Defenses in Text"

Unfortunately, studies that address the defense mechanisms and robustness of the model are few and generally revolve around gradient-based training. Moreover, few studies deal with adversarial training of BERT [?, ?].

In this master thesis report, a different BERT model fine tuning approach which can create result in comparatively robust model without compromising with original accuracy. Moving forward, conducting experiment to observe the performance both models created by proposed and conventional approach under attack and not under attack situation. The proposed training methodology is based on a semi-supervised approach called Mean Teacher proposed by Tarvarian et. al [?]. They propose to train two identical model called student and teacher with two different training methods. Their research indicates the teacher model to be more robust. This approach performed well in speech recognition and image processing tasks, but performance in NLP tasks was an open scope for experiment and different data augmentation specific to NLP domain is proposed. In the mean teacher approach , teacher model is utilized as a predictor, and utilized synthetic prominent adversarial unlabel data instead of unlabeled data.

There are many factors that motivated to perform this experiment. One, to study the performance of language models under worse situation so mitigation strategies can be planned. Till now, no study has been conducted which discuss the language model behaviour under attack and proposing a defensive fine tuning mechanism. Second, study the performance of different attack techniques to know their strength and weaknesses, or if possible any pattern, it always better to know your attacker. Finally, proposing an approach for robustness of language as defensive strategies. *Different researchers worked tirelessly and showed that DNN models were vulnerable in object recognition systems (Goodfellow et al., 2014), audio recognition(Carlini and Wagner, 2018), malware detection (Grosse et al., 2017), and sentiment analysis systems (Ebrahimi et al., 2017) as well. An example of the adversarial attacks is shown. [**?**] In the field of NLP, Papernot (2016) paved the way by showing that adversarial attacks can be implemented for textual data as well. [**?**]*

TODO: What are the research question ?

Research Questions:

1. How does BERT model works and understanding the Transformer architecture?
2. Empirical study of performance proposed model and BERT model in terms of efficacy in absence of adversarial attacks.
3. Empirical study of performance of proposed model and BERT model under adversarial attacks.

TODO: As a result of experiment, found that Evaluated with what and dataset details ?
BERT attacking BERT, PWWS, TextBugger, Textfooler.
Evaluation Results
short conclusion

2 Background (28-30 pages)

2.1 Natural Language Processing (2 pages)

- What is Natural language processing? (1 paragraph)
- What is Text classification problem and how it is solved ? (1-2 paragraph)
- Recent advancements in Natural Language Processing related to text classification.(1-2 paragraph)

2.1.1 Data Representation (2 pages)

- What is Data Representations Or How machine Understand data ? (1 paragraph)
- How text data is represented? (1-2 paragraph)
- Recent advancements in Data Representation.(1-2 paragraph)

2.2 BERT(Bidirectional Encoder Representation From Transformers)

BERT(Bidirectional Encoder Representations from Transformers) was proposed by Devlin et. al [**?**], mainly based on Transformers [**?**], but not limited to it. BERT is basically a transformer encoder stack that outputs the representation context also called pre-trained model. BERT model is pre-trained on deep bidirectional representation of large unlabeled text in both right and left context, which can be trained further called fine-tuning by ending additional output layers to get state-of-art result in various NLP tasks like text classification, question answering, language inference, language translation and so on.

The main advantage of BERT based is simplifying the process of NLP tasks in machine learning and open access to contextualized embedding trained huge amount of words which is quite impossible at individually. Unfortunately, it is highly computational intensive which makes it costly at production scale and demands high performance computational machine. In order to understand, how actually BERT model works , we need to initially understand the transformer attention mechanism and then BERT model. Hence, the intention behind next section is clear.

2.3 Understanding Transformers Architecture (11 pages)

Previously, NLP tasks were solely depends on sequential model like CNN, RNN, LSTM and BiLSTM models which has disadvantage of computational expensive, lack of distributing capabilities, and only satisfactory performance. In December 2017, Vaswani et. al [**?**], proposed a simple architecture is based attention mechanism called the transformer architecture which outperformed the existing state-of-art NLP models shown in figure 2.1 . This proposed model architecture comparatively can be trained faster and showed better evaluation result. This transformer model is a revolutionised and game changer architecture for Natural Language Understanding(NLU) and Natural Language Processing (NLP). Moving forward, became one of the main principle behind recent break through and state of the art language models like BERT, GPT, and T5.

The transformer is solely based on a special type of attention mechanism called *self attention*

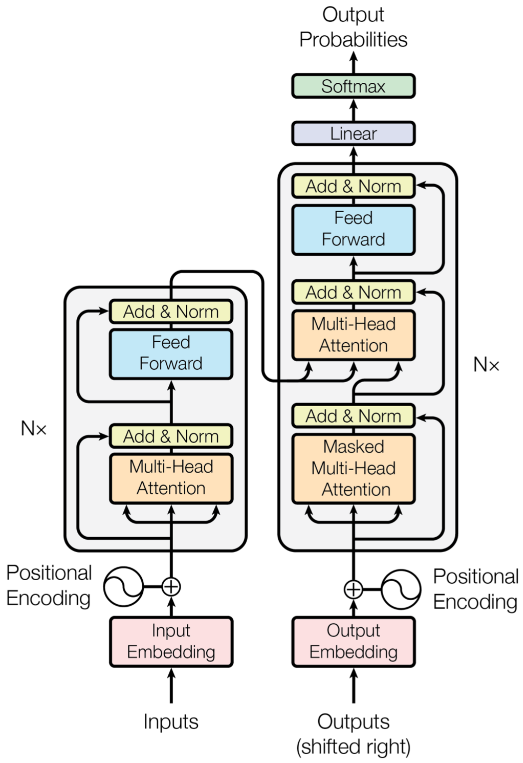


Figure 2.1: Transformer Model Architecture [**?**].

and completely gets rid of recurrence. Transformer is an encoder and decoder stack where the encoder reads the inputs and outputs a representation as a context vector also called as *contextualized embedding* as shown in figure 2.2, based on single-head attention or multi-head attention, and the decoder makes predictions based on those context vectors. In proposed architecture by Vaswani et. al [**?**], transformer model is a composed of 6 layers of encoder stacked on each other and same applies to decoders. Each encoder is composed of multi-head attentions followed by layer normalization and Feed forward network and the only difference in Decoder is before multi-head attentions it has masked multi-head attentions layer.

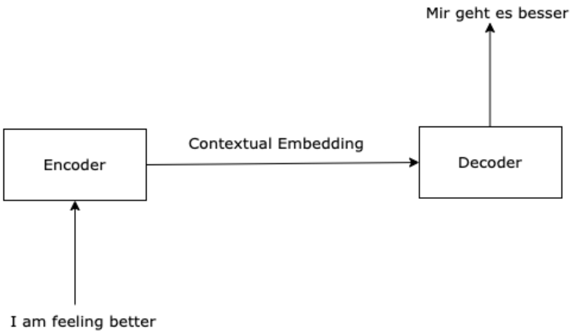


Figure 2.2: Transformer Encoder Decoder.

2.3.1 Encoder and Decoder (2 pages)

- What is Encoder and Decoder ? (1 paragraph)
- Working of Encoder and Decoder architecture in context of Transformers. (1-2 paragraph)

2.3.2 Self Attention Mechanism (3 pages)

- What is self attention Mechanism. (1-2 paragraph)
- Working of Self Attention Mechanism in context of Transformers.(3-4 paragraph)

2.3.3 Multi-head Attention Mechanism (2 pages)

- What is Multi Head attention Mechanism. (1 paragraph)
- Working of Multi Head Attention Mechanism.(1-2 paragraph)

2.3.4 Positional Encoding (2 pages)

- What is Positional Encoding? (1 paragraph)
- Working of Positional Encoding in context of Transformers. (1 paragraph)

Working of Transformers (1-2 pages)

- Now Combining all methods and explaining Transformers architecture ? (1-2 paragraph)

2.3.5 Decoder

2.4 Training and Fine-tuning of BERT (3 pages)

BERT(Bidirectional Encoder Representations from Transformers) was proposed by Devlin et. al [**?**], mainly based on Transformers [**?**], ULMFit [**?**], ELMo [**?**], and the OpenAI transformer [**?**] but not limited to it. The transformer is an encoder and decoder in which the encoder reads the inputs and outputs a representation as a context vector, based on single-head attention or multi-head attention, and the decoder makes predictions based on those context vectors. However, BERT is the only Transformer Encoder stack that outputs the representation context. Moreover, unlike OpenAI transformers, which read data from left to right or right to left, BERT reads complete sequences at a time, making it bidirectional. For training the large amount of unlabelled data, the main challenge is the lack of a label or a goal, so BERT uses two different learning strategies called Masked Language Model(MLM) and Next Sentence Prediction(NSP). In MLM, 15% of words are replaced with [MASK] tokens, and the BERT model predicts the masked word based on other words in the sequence.
In NSP, BERT model is given two pairs of sentences with [CLS] as the sentence start and [SEP] as the separation between sentences. Then, BERT model predicts whether the next sentence is the correct one or random. BERT model is pre-trained on a large amount of unlabeled text, but still, fine-tuning is required for specific tasks.
A BERT paper [**?**] described two BERT models on which they conducted their experiments.

1. $BERT_{BASE}$: 12 Transformers blocks(Encoder, L), 768 Hidden Units(H), Attention Heads(A) 12, Total Parameters 110M.

2. $BERT_{LARGE}$: 24 Transformers blocks(Encoder, L), 1024 Hidden Units(H), Attention Heads A 16, Total Parameters 340M.

DistilBERT, is another more compact version of BERT proposed by Victor et. al [**?**], is comparable to BERT. Both the proposed model is susceptible to adversarial attack, and few studies have examined adversarial training of these language models or the performance against word-level attacks.

2.5 Adversarial Attacks (3 pages)

On the other hand, it is showed that robustness and generalization of ML models can be improved by crafting high-quality adver- saries and including them in the training data (Goodfellow, Shlens, and Szegedy 2015) textfooler. *In the image domain, the perturbation can often be made virtually imperceptible to human perception, causing humans and state-of-the-art models* *to disagree. However, in the text domain, small perturbations are usually clearly per- ceptible, and the replacement of a single word may drastically alter the semantics of the sentence. TEXTBUGGER*

2.5.1 Types of Adversarial attacks

Under the black-box setting, the attacker is not aware of the model architecture, parameters, or training data. It can only query the target model with supplied inputs, getting as results the predictions and corresponding confidence scores. Under the black-box setting, gradients of the model are not directly available, and we need to change the input sequences directly without the guidance of gradients(TextBugger).

2.5.2 Limitation And Constraints

A major bottleneck in applying gradient based (Goodfellow et al., 2015) or generator model (Zhao et al., 2018) based approaches to generate adversarial examples in NLP is the backward propagation of the perturbations from the continuous embedding space to the discrete token space. [**?**] *For instance, adversarial text detection is only suitable for certain adversarial attacks. Model enhancement like adversarial training suffers the shortcoming in distinguishing adversarial texts generated by unknown adversarial techniques."Towards a Robust Deep Neural Network in Texts: A Survey"*

2.5.3 Different attack methodology in Text Classification problem

3 Related Work(2 pages)

- Mentioning the related work and their results. (3-4 paragraph)
- *At present, adversarial texts detection [24] and model enhancement [13] are two mainstream ideas in fighting against the threats of adversarial texts, but both of them exhibit obvious weakness."Towards a Robust Deep Neural Network in Texts: A Survey"*

*Belinkov (2017) in their experiments showed that training the model with different types of mixed noises improves the modelâs robustness to different kinds of noises "Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation.". In the experiments of Li (2018) they also showed for TEXTBUGGER at- tack adversarial training can improve model per- formance and robustness against adversarial examples. In the experiments of Zang (2019) they showed that their sememe based substitution and PSO based optimization improved classifiersâ ro- bustness to attacks. By using CharSwap during ad- versarial training on their attack Micheal showed that adversarial training can also improve the ro- bustness of the model "Textual adversarial attack as combinatorial optimization". Till now no defense strategy can handle all different types of attacks that were mentioned here. Each defense strategy worked on a sin- gle type of attack approach. For example, for spelling mistakes, we can use the defense technique proposed by (Pruthi et al., 2019). For synonym based attacks we can use the SEM model. A unified model that can tackle all these issues has not been proposed yet. [**?**] Overfitting may be another reason why adversarial train- ing method is not always useful and may be only effective on its corresponding attack. This has been confirmed by Tramâer et al. [76] in image domain, but it remains to be demonstrated in text."A survey on Adversarial Attacks and Defenses in Text"*

One of the related research paper proposed by Li et. al [**?**], in their proposed approach, the BERT model is used to generate word replacements for the target word. First, they identify the most important words of the BERT model i.e. the words in the sequence have a high significance influence on the final output logit. Then, by using another BERT model, they try to replace these words with the target word by utilizing its MLM capability. As per their claim, the average after attack accuracy was lower than 10% and perturb percentage was less than 10%. However, during the process of generation, there are chances of compromising with semantic constraints.

There is tool available like TextAttack, which has the capability to generate grammatically correct sentences using semantic constraints and also provide 16 various important attack framework based on recent research which can be used as baseline. So, this tool can be adapted for producing adversarial datasets for training and can also be used to test the effectiveness of proposed model.

TextDeceptor [?] proposed a text attack approach, first they rank sentences and words and then replace them with similar words based on cosine similarity between word vectors, also considering POS (part-of-speech), which helps them to get grammatical correctness. In addition, this approach can be employed to generate adversarial text for the proposed master thesis topic.

Yankun et. al [?] proposed an approach that generates real-world meaningful text automatically using a variational encoder and decoder model, however, the sentences are sometimes completely different than the original.

Sun et. all [?] proposed research on generating adversarial misspelling and observing the performance of BERT. It was found that the BERT model is prone to misspelling attacks and accuracy drops by 22.6 % on Stanford Sentiment Treebank(SST) dataset.

Word level Textual Adversarial attack proposed by Yuan et. al [?], they are using sememe based word substitution. Using sememe-based word substitution is supposed to be more accurate since the substituted word has probably retained the same meaning. As per their claim, their attacks are notably having 98.70% success rate on IMDB dataset.

Siddhant et. al [?] proposed BERT's masked language models to generate alternate words for masked tokens, possible adversarial examples are derived. textattack tool has included respective approach in the package.

Research related to adversarial training of language model is few. Liu et. al [?] BERT model requires considerable computational power to perform virtual adversarial training [?] during pre-training. Due to memory constraint, pre-training BERT model is out of scope.

My understanding is that a more recent and closely related approach is proposed by Danqinq et. al [?], where adversarial training is done by fast gradient methods [?] and ensemble methods where multi-BERT model prediction aids in achieving robustness. The proposed approach is also gradient-based and uses multiple BERTs for prediction, which raises concerns about computation.

4 Proposed Methodology (2 pages)

- Working of Proposed Methodology. (1-2 paragraph)
- Discuss Proposed Research Questions. (2-3 paragraph)

In general, existing attack algorithms designed for images cannot be directly applied to text, and we need to study new attack techniques and corresponding defenses. TextBugger The proposed method calls for fine-tuning the BERT model using the mean-teaching approach for classification tasks and the adversarial unlabeled dataset for training. As far as I know, this mechanism has not been examined. Due to memory constraints, I prefered focusing on fine-tuning the BERT model rather than pre-training. Pictorial representation for reference is shown in figure 4.1.

The adversarial unlabeled data can be generated using recently available tools like textattack [**?**], which generate semantically correct adversarial text, inter-class most important word exchange which shares the same meaning, and back translation methods. This algorithm relies on adversarial texts, which are prominent texts that can affect model performance. Since these texts are derived using label data, robustness may be achieved by learning more representations.

Proposed approach is using Classification cost($C(\theta)$) is calculated as binary cross-entropy as mentioned in Mean teacher paper. However, Consistency cost($J(\theta)$) is mean squared difference between the predicted outputs of student with adversarial unlabeled data (weights θ, adversarial data x_{adv}) and teacher model (weights $\hat{\theta}$,x_{adv}). And, KL divergence loss can be another option. The mathematical declaration is as follows.

$$J(\theta) = \mathbb{E}_{x_{adv}}[\|f(x_{adv},\theta) - f(x_{adv},\hat{\theta})\|^2] \tag{4.1}$$

While back propagating in student model, the overall cost ($O(\theta)$) **??** and exponential moving average **??** is same as mean teacher approach. However, alpha and ratio will be tuned as per our requirement and performance. Unlike mean teacher approach in computer vision, adding noise strategy is quite different in Natural Language Processing. Random noise represents unknown words during training, which can affect model performance. Hence, instead of adding noise to labeled data, significant adversarial data is employed to increase robustness is proposed for experiment which could play the role of regularization.

As far as I am aware, no specific defense mechanism is tied to our proposal to build a BERT model in mean teacher fashion in order to ensure robustness in classification tasks requiring

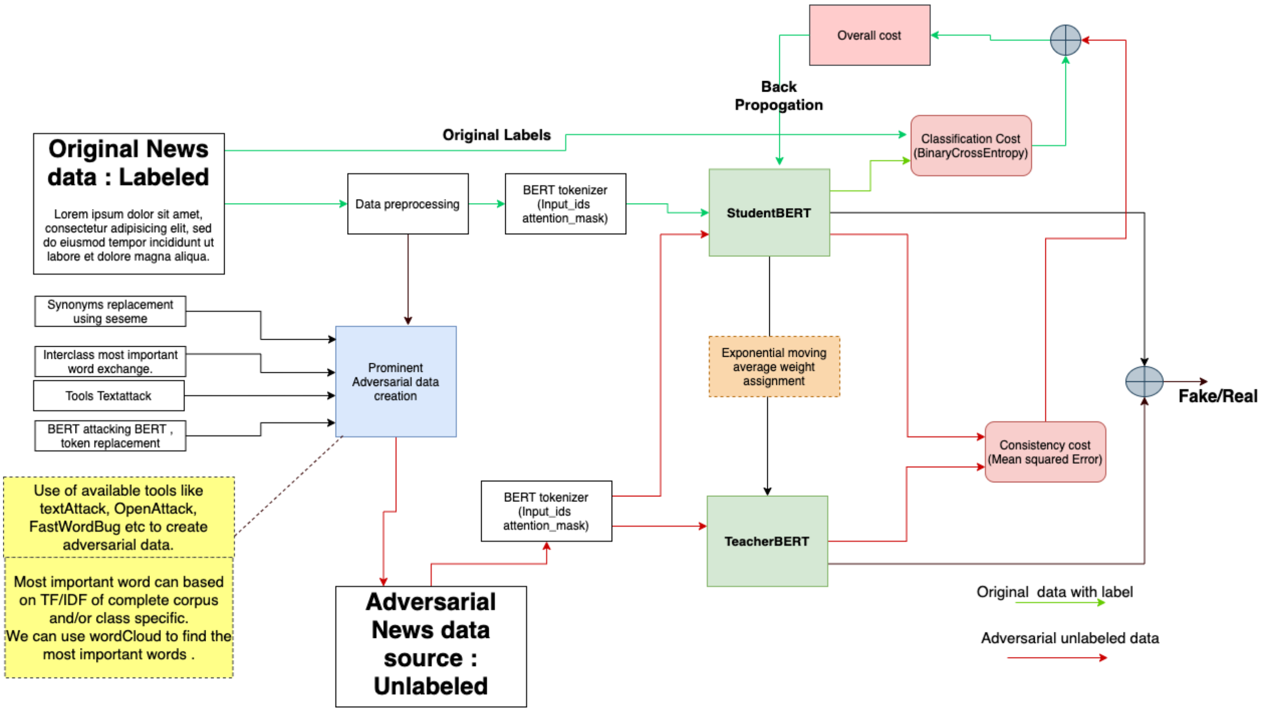

Overall cost
Back Propogation
Original Labels
Classification Cost (BinaryCrossEntropy)
Original News data : Labeled
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Data preprocessing
BERT tokenizer (Input_ids attention_mask)
StudentBERT
Synonyms replacement using seseme
Interclass most important word exchange.
Tools Textattack
BERT attacking BERT , token replacement
Prominent Adversarial data creation
Exponential moving average weight assignment
Fake/Real
Consistency cost (Mean squared Error)
Use of available tools like textAttack, OpenAttack, FastWordBug etc to create adversarial data.
Most important word can based on TF/IDF of complete corpus and/or class specific. We can use wordCloud to find the most important words .
BERT tokenizer (Input_ids attention_mask)
TeacherBERT
Adversarial News data source : Unlabeled
Original data with label
Adversarial unlabeled data

Figure 4.1: Proposed methodology

relatively few computing resources and being relatively simple. Also, the proposed approach does not require pre-training the BERT model. Therefore, there is the possibility of studying the performance of the proposed training methodology by utilizing relevant word-level attack approaches and observing the performance. In the proposed thesis, the focus of study would be :

1. Observing the robustness of the individual BERT model and the mean teacher BERT model without adversarial training (in this case, instead of adversarial unlabeled data, I will be using labeled data with dropout).

2. Observing the robustness of the Mean Teacher BERT model with adversarial training.

3. Observing the effectiveness of available attacks tools after adversarial training with Mean Teacher BERT.

4.1 Research Questions (3 pages)

- Discussion regarding Research question in context of proposed methodology.

5 Experiment(7 pages)

5.1 Dataset (1-2 paragraph)

For assessing the performance of baseline model and proposed model. We have selected two datasets, one is Covid-19 fake news dataset provided at Codalab competition and IMDB review binary classification dataset. IMDB dataset is sentiment classification dataset which contains movie reviews of the user and having two classes positive and negative and most generally used in classification and adversarial attack research papers. This dataset has 50,000 labelled but due to computational limitation, we have filtered and sampled only dataset whose length is in between 6 to 150 as augmentation process takes quite longer time which leads 6000 samples to train and evaluate our model. The train and test size is show in table 5.1. And, we have sampled 6000 training labeled samples to create augmented unlabeled dataset. Average length of the filtered dataset is 100. The label distribution is completely balanced in training dataset.

Covid-19 Fake news dataset is recent dataset specific for fake news detection tweets related to COVID 19 with label as fake and real. Covid-19 fake news dataset size is 8000 and we have completely utilized this dataset. Observing the performance of proposed model in more recent fake news dataset is motivation behind selecting the dataset. In contrast to IMDB dataset, the average length of Covid-19 fake news dataset is 25 as mentioned in table 5.2. Covid-19 fake news dataset has mostly hashtags and less English words vocabulary which might create challenge for those recipes, we would like to investigate the performance of models under this scenario too. The label distribution is almost balanced as compared to IMDB dataset which is completely balanced. The intention is to observe the effect of label distribution.

Dataset	Train	Test	Unlabeled	Aug. Unlabeled
codalab (Positive/Negative)	3199/2891	1071/969	xxxx	xxxx
IMDB (Fake/Real)	xxxx/yyyy	xxxx/yyyy	xxxx	xxxx

Table 5.1: Train/ Test split details of dataset

5.2 Data Pre-processing and Exploration (3 pages)

As language models are based on learning the context of the sentences hence least affected by stop words and removing those words might affect the performance. Hence, the one of the benefit of language model is its negligible requirement of data cleaning or no data cleaning. In our case we have performed following data pre-processing steps:

1. HTML tags removal.
2. Digit removal.
3. Lower casing.
4. Punctuation removal.

For achieving the particular task, we have utilized texthero python library which provide function related to data pre-processing and exploration.

Data Exploration (3 pages)

For training the mentioned models, we selected almost equal distribution of label in training data and test data, and same training data we have utilized to create unlabeled augmented data for training via proposed method as shown in 5.1.

Dataset	Avg. Length
codalab	xxxx
IMDB	xxxx

Table 5.2: Train/ Test split details of dataset

5.3 Data Augmentation

For creating the unlabeled augmented dataset, we have utilized three strategies :

1. Synonym Augmentation
2. Context Based Augmentation
3. Back translation.

The reason behind calling this dataset unlabeled augmented dataset is while augmenting the dataset there are high chance that the information is changed or completely opposite in contrast to label. Therefore, once we augment the data , we will not be using the label of augmented data hence unlabeled augmented dataset. During experiments, various python packages like text attack , nlpaug, and various basic python packages for synonym changes we tried. But, in the end , considering the time and computation constraint, we have selected nlpaug data augmentation package to achieve all three augmentation strategies and can be accessed in following link. To augment the dataset, we have taken train dataset with dropping the label

columns , then we split this dataset into three part. One part for synonym augmentation, second part for context based augmentation, and remaining for back translation.
For synonym augmentation, wordNet lexical English database is used as augmentation source which consist of word definitions, hyponyms, and semantic relationship. Same database in our case utilized for synonym replacement. Parameter maximum augmentation(aug_max) is used to control the level of augmentation. It is set to 50 and 15 for IMDB and Covid-19 fake news dataset respectively. One more parameter called *iter* is utilized to create two different copies of synonym augmentation dataset.
TODO: Image of synonym augmentation.
Context based augmentation is based on replacing the words the words in the sentence without changing the context. Generally, language models are used to achieve this particular task, hence it is quite time and memory expensive. Therefore, DistilBert language model is utilized to perform this augmentation.
Back translation is the process of converting sentences in different languages and then translating back to original language. Like, context based augmentation, Marian translation framework is utilized , hence time and memory expensive. In our experiment, we are converting sentence into Romance language and back to English. We have used CITE model to perform this experiments. Marian translation framework is comparatively an free, faster and efficient .
TODO: Image for Back translation.

5.4 Experiment Environment description (1-2 paragraph)

To successfully perform experiments, Google colab notebook with GPU is utilized to perform the experiments. TODO: GPU details need to mention or image.

5.4.1 Hyper parameter Details (1 paragraph)

As shown in table 5.3, to train the baseline model and proposed model, we have used these parameter values. However, observing the performance with different settings is not the current focus and open for future task.

5.4.2 Model architecture (2-3 paragraph)

TODO: Image of model architecture

5.5 Metrics (2 pages)

- Definitions of Metrics used for evaluation. (2-3 paragraph)
- Metrics are :

Hyper parameter	Used parameters in this work
Optimizer	Adam
Learning rate	$2\epsilon - 5$
Loss function	Binary Cross Entropy
Epochs	3
Batch Size	4
Loss Ratio	0.5
Alpha	0.99
Dropout	0.2
Max length	100

Table 5.3: Hyper-parameters Details

- Original Accuracy
- Accuracy under attack
- Attack success rate
- Average perturbations word
- Average number word per input
- Average number of queries

5.6 Threat Model

- Defining about the level of information is exposed.
- Assumptions
- Threats

5.7 Text Attack Recipes and Tool (2 pages)

To evaluate the proposed approach, four black box attack recipes has been selected which satisfy lexical, grammatical, and semantic constraints. To utilize all the attack recipes to evaluate baseline BERT model and proposed Mean Teacher BERT, we have TextAttack python package [**?**]. is In this section, we will discuss about their attacking principle, working and characteristics.

5.7.1 TextFooler

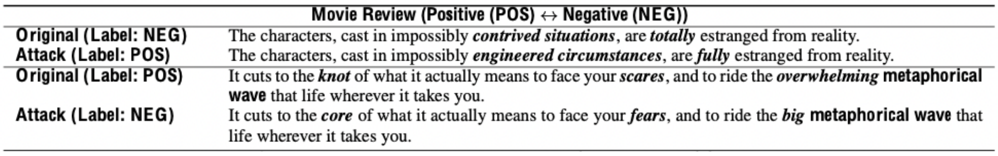
Di jin et. al. proposed Textfooler [**?**], a simple and effective adversarial attack generation strategy in black box settings which has characteristic of preserving the semantics, and grammar which they called utility-preserving adversarial examples as shown in figure 5.1. For better understanding, we briefly explain the three steps process of generating adversarial attacks below:

1. **Word Importance Ranking**: Given sentence of words, they create ranking of each word by calculating change before and after deleting the words called importance score. Followed by filtering out stop words using NLTK and spaCy libraries just to preserve the grammar of the sentence.

2. **Word Transformer**: To replace the word with synonym, they have utilized novel word embeddings proposed by MrksËic Ì et. al. [**?**] which is basically injects antonymy and synonymy into vector space representations to improve vectors capability of semantic similarity. The replacement policy completely depends on three constraint (1) Similar semantic similarity, (2) Fit within the surrounding context , (3) attacks the model. To calculate the similarity, using Universal Sentence Encoder proposed by Cer et al. [**?**], for encoding the sentence into high dimensional vector and calculating the cosine similarity between sentences. Then, selecting replacement candidates which has value above preset threshold value and create a pool of candidate.

3. **Replacement**: Among pool of candidates, if there already exist any candidate that can alter the prediction of target model then candidate with highest cosine similarity score between original and adversarial sentence is selecting. Otherwise, lower confidence score of label is selected.

TextFooler, has accessed the performance of BERT model under adversarial attack using IMDB movie review dataset . As per their experiment, the accuracy significantly dropped from 90.9% to 13.6 % with perturbed words 6.1 , number of queries sent to target model 1134 and average length of the IMDB dataset 215. Furthermore, TextFooler is computationally inexpensive and complexity increases linearly with respect to text length.



	Movie Review (Positive (POS) ↔ Negative (NEG))
Original (Label: NEG)	The characters, cast in impossibly ***contrived situations***, are ***totally*** estranged from reality.
Attack (Label: POS)	The characters, cast in impossibly ***engineered circumstances***, are ***fully*** estranged from reality.
Original (Label: POS)	It cuts to the ***knot*** of what it actually means to face your ***scares***, and to ride the ***overwhelming*** **metaphorical wave** that life wherever it takes you.
Attack (Label: NEG)	It cuts to the ***core*** of what it actually means to face your ***fears***, and to ride the ***big*** **metaphorical wave** that life wherever it takes you.

Figure 5.1: TextFooler example [**?**]

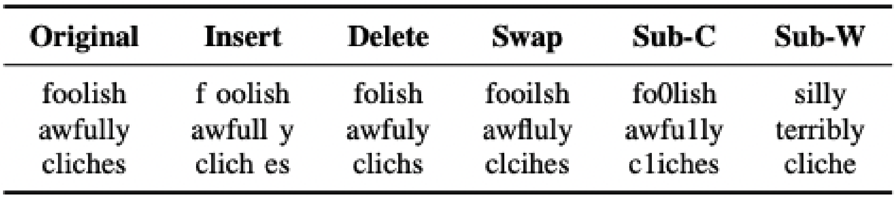
5.7.2 TextBugger

TextBugger proposed by Jinfeng Li et. al. [**?**], is based on misspelling of words or characters which are visually and semantically similar to the original text for human being. A simple misspelling can lead token to 'Unknown' which is mapped to unknown tokens id can also force machine learning model to behave incorrectly. On the other hand, studies shows that similar misspelling can still be perceptibly or inferred by the reader [**?**, **?**] . This attack is focused on both character-level and word-level perturbation. Jinfeng Li et. al has proposed both white box and black box attack generation strategies, however, our report is focused on black box attack. Black box attack generation strategies is briefly discussed in Three steps:

1. **Finding Important Sentences**: The importance score of individual sentences in an article is determined by confidence score of particular sentence by target model.

2. **Finding Important Words**: The importance score of word is the difference between confidence of target model with word and without word.

3. **Bugs Generation**: In TextBugger, they use five bugs generation strategy (1) **Insert**: Inserting space into words, (2) **Delete**: Deleting random character, (3) **Swap**: Swapping random adjacent character, (4) **Substitute-C**: Substitute character with visually similar characters, and (5) **Substitute-W** : Replacing word with top-k nearest neighbour in context aware word vector space , as shown in figure **??**



Original	Insert	Delete	Swap	Sub-C	Sub-W
foolish	f oolish	folish	fooilsh	fo0lish	silly
awfully	awfull y	awfuly	awfluly	awfu1ly	terribly
cliches	clich es	clichs	clcihes	c1iches	cliche

Figure 5.2: TextBugger 5 bug generation strategies [**?**]

TextBugger model is evaluated against LR, CNN, and LSTM using IMDB movie review dataset, and have shown 95.2%, 90.5% and 86.7% respectively, with perturbed word 4.9%, 4.2% and 6.9 % respectively. However, observing the effectiveness against BERT model is still not evaluated and is explored in this report. Unlike TextFooler, TextBugger computational complexity is sub-linear to text length and can generate adversarial attacks in comparatively less time.

5.7.3 Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency (PWWS)

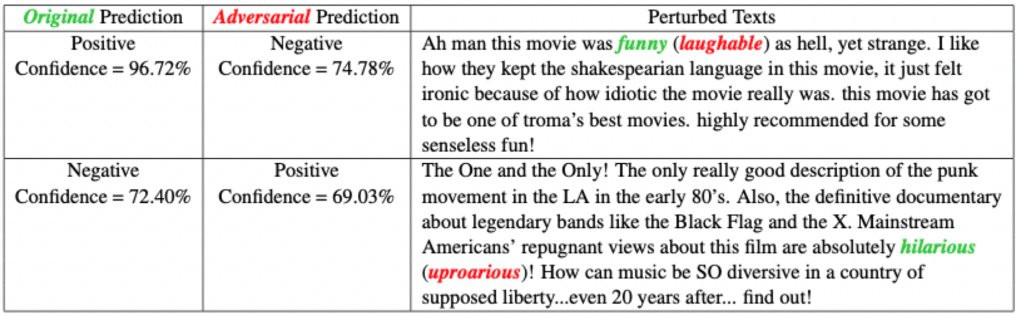
Shuhuai et al. [**?**] proposed method of synonym and named entity (NE) replacement method which is determined by the words saliency and the classification probability, and proposed a greedy algorithm called probability weighted word saliency(PWWS). To replace the word with the synonym is decided by significant change in classification probability and at the same time have minimum word saliency. Their approach is mainly can be explained in two steps as follows:

1. **Word Selection Strategy**: Calculating word saliency vector of each word in a text, and prioritizing the words according to degree of change in classification probability after

replacement as well as minimum word saliency of that words. Here, word saliency defined as degree of change in classification probability of the model if the word is set to unknown [**?**].

2. **Replacement Strategy**: To find the substitution, they used WordNet to find the synonym of the words. And, if word is Named Entity(NE), then replacing the NE with similar type NE appeared in opposite class.

Finally, greedily iterate through words replacement to make model change the label. This approach has been evaluated with Word based CNN [**?**], Bi-directional LSTM model, LSTM and Char-based CNN [**?**], however, still have open scope for evaluating against language models. Considering Bi-LSTM result , the accuracy dropped from 84.86 % to 2.00% with perturbation 3.38% for IMDB dataset, example is shown in figure 5.3. However, the computational and time complexity of the proposed approach is comparatively higher than other discussed strategies.



Original Prediction	***Adversarial*** Prediction	Perturbed Texts
Positive Confidence = 96.72%	Negative Confidence = 74.78%	Ah man this movie was ***funny*** (***laughable***) as hell, yet strange. I like how they kept the shakespearian language in this movie, it just felt ironic because of how idiotic the movie really was. this movie has got to be one of troma's best movies. highly recommended for some senseless fun!
Negative Confidence = 72.40%	Positive Confidence = 69.03%	The One and the Only! The only really good description of the punk movement in the LA in the early 80's. Also, the definitive documentary about legendary bands like the Black Flag and the X. Mainstream Americans' repugnant views about this film are absolutely ***hilarious*** (***uproarious***)! How can music be SO diversive in a country of supposed liberty...even 20 years after... find out!

Figure 5.3: Example attack of PWWS [**?**]

5.7.4 BAE: BERT-Based Adversarial Examples

Garg et al. [**?**] proposed a novel black box approach to generating adversarial examples by utilizing BERT masked language model(MLM). According to their proposed approach, first they calculate words importance by computing the decrease in probability of predicting the correct label after deleting that particular word, similar to Textfooler [**?**] and PWWS [**?**]. And, using pre-trained BERT MLM model, where a particular word is replaced with MASK token and let the MLM model predict the context specific words. Then, filter top K tokens based on most similarity score(Threshold 0.8) using Universal Sentence Encoder [**?**] and removing words that doesn't fall into similar part-of-speech(POS) as the original word. Now, replacing the original word with top K(50) tokens, iterate from most similar token in decreasing order until attack is successful and trying all combination. A schematic working diagram is shown in the figure 5.4.

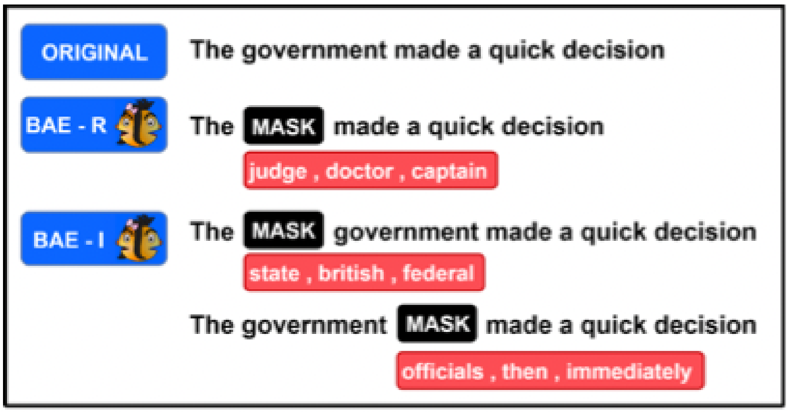

ORIGINAL
The government made a quick decision
BAE - R
The MASK made a quick decision
judge , doctor , captain
BAE - I
The MASK government made a quick decision
state , british , federal
The government MASK made a quick decision
officials , then , immediately

Figure 5.4: Schematic working and example of BAE [**?**]

6 Result Analysis (3 pages)

Values in the table are dummy values

- Tabulations related to result observed during experiments. Table
- Model performance under different attacks. (1-2 paragraph)
- Comparison of baseline and proposed model.(1-2 paragraph)

Text bugger types of attack is not added in augmented unlabeled data which might effect the performance of the model based . Codalab length and most words are hashtags, can lead to change in the performance in models.

TextFooler, a strong black-box attack baseline for text classification models. However, the adversarial examples generated by TextFooler solely account for the token level similarity via word embeddings, and not the overall sentence semantics. This can lead to out-of-context and unnaturally complex replacements (see Table 3), which are easily human-identifiable. Consider a simple example: âThe restaurant service was poorâ. To- ken level synonym replacement of âpoorâ may lead to an inappropriate choice such as âbrokeâ, while a context-aware choice such as âterribleâ leads to better retention of semantics and grammaticality BAE. Evaluation as per text length.

Attack Recipe	Model	Acc. und Attack(%)	Acc. Succ. Rate(%)	Avg. Pert. Word(%)	Avg. No. Queries	Ori. Acc.(%)
BAE	BERT	33.93	63.77	3.78	242.24	93.67
	DistilBERT	33.25	64.18	3.56	238.20	92.80
	MT BERT	56.45	40.03	3.55	198.26	94.13
	MT DistilBERT	53.50	42.51	3.31	285.70	93.05
PWWS	BERT	0.60	99.36	3.97	749.33	93.67
	DistilBERT	1.70	98.17	3.98	750.12	92.80
	MT BERT	23.20	75.35	5.70	890.84	94.13
	MT DistilBERT	17.55	81.14	5.37	867.65	93.05
TextBugger	BERT	2.30	97.54	22.04	235.27	93.67
	DistilBERT	5.45	94.03	20.80	258.47	92.80
	MT BERT	35.13	62.68	28.57	449.96	94.13
	MT DistilBERT	30.05	67.70	26.66	420.39	93.05
TextFooler	BERT	0.10	99.89	5.14	279.12	93.67
	DistilBERT	1.07	98.85	5.07	278.73	92.85
	MT BERT	30.92	67.16	8.04	720.77	94.13
	MT DistilBERT	25.48	72.64	7.54	613.91	93.17

Table 6.1: Textbugger Experiment Result

7 Conclusion and Future Works (3 pages)

- Answering the research questions here. (3-4 paragraph)

7.1 Limitations (1 page)

- What Problem and limitations observed during experiments. (1-2 paragraph)
- Problem and limitations of the proposed methodologies. (1-2 paragraph)

Challenges:

- Computational challenges and time limitation. (Proposed approach is computationally expensive than baseline model)
- Text attack challenges

7.2 Future Work (1 page)

- Future work and improvements.(1-2 paragraph)
- Evaluation of the model with different length text and performance of adversarial attack has open scope of work.
- Effectiveness of different augmentation techniques like back translation, context augmentation, and synonym can be evaluated.
- Effectiveness's of vast amount of unlabeled data can be utilized in the future instead of utilizing the train data.
- Proposed approach can still be evaluated with recent state of the art language model.
- Including other types of adversarial example in augmented data like TextBugger.
- Including adversarial dataset using attack recipes can be evaluated in the future. Like PWWS claims that including their adversarial training data can increase the robustness of the model. Only, challenge that come in mind in current situation is highly time and computationally expensive.

7.3 Conclusion (1 pages)

- Summary of the master thesis experiment.(1-2 paragraph)

References