Localization of Fake News Detection Via Multitask Transfer Learning

A Thesis Proposal
Presented to
the Faculty of the College of Computer Studies
De La Salle University Manila

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science by

CRUZ, Jan Christian Blaise B.
TAN, Julianne Agatha M.

Charibeth K. CHENG
Adviser

July 17, 2019

Abstract

The capability of the internet and social media to quickly spread fake news reinforces the need for computational tools to be made to combat its spread. While techniques exist to train classifiers that detect fake news, they assume the abundance of resources such as large labeled datasets and expert-curated corpora, which low-resource languages do not have. In this paper, we show that transfer learning techniques can be used to train fake news classifiers from little data. We achieve 91% accuracy on a fake news dataset in the low-resourced Filipino language, reducing the error by 14% compared to few-shot baselines. Furthermore, we propose Stylometric Multitask Transfer, a novel multitask learning-inspired technique that allows transfer learning methods to adapt to the stylometry of its target task while being finetuned for classification. Using this, we improve transfer learning performance by 4-6%, with our best performing model achieving an accuracy of 96%. Lastly, we also show that our method generalizes well to different types of news articles, including political news, showbusiness news, and opinion articles

Keywords: Few-Shot Learning, Natural Language Processing, Classification, Localization

Contents

List of Figures

List of Tables

Chapter 1

Research Description

This chapter contains an analysis of current state-of-the-art works on fake news detection using various methods, as well as their drawbacks that prevent successful localization into low-resource languages. This will provide context and background on the proposed research. Techniques such as few-shot and transfer learning, as well as applications related to the research are outlined and reviewed. Research objectives, scope, limitations, and significance will also be outlined within the chapter.

1.1 Current State of Technology

News consumption is continuously shifting away from traditional formats like print and broadcast into a more electronic medium of exchange, one of which is social media. In a study by the Pew Research Center in 2017, it is found that 67% of Americans consume their news on social media, with sites like Twitter, YouTube, and Snapchat seeing an 11 to 15% increase in news usership since 2013 (*Journorg:2017*, n.d.-a). The reasons for this shift are deeply rooted in the nature of these online platforms. News on social media is often more timely and less expensive to access compared to traditional mediums. In addition, it is easier to share news obtained on social media with other readers, such as friends and immediate connections (Shu et al., 2017).

Due to the inherent advantages that social media offers, false and unreliable news are able to spread easily and quickly. One study found that false news spreads faster than true news on Twitter, with the top 1% of around 126,000 rumors reaching 1000 to 100,000 users as opposed 1 to true news which rarely

reached the 1000 user mark (Vosoughi et al., 2018). This quick-spread is hard to control, and causes damage that is equally hard to reverse. Specifically, people find it hard to adjust their judgment if the information that formed it are later proven to be false (Dekeersmaecker & Roets, 2017).

In response to the increasing need to then validate the news people consume, there has been a growing interest in research revolving around automated fake news detection and fact-checking (Pérez-Rosas et al., 2017). These efforts also include the development of tools to automatically discern if a certain news article or publication is fake. Such tools are commonly referred to as fact-checking tools.

1.1.1 Fact Checking Tools

In order to combat the rapid spread of fake and unreliable news, various organizations have created fact-checking tools. Existing tools for fact-checking articles and publications can be categorized into three groups, namely expert-oriented, crowdsourcing-oriented, and computational-oriented (Shu et al., 2017).

Expert-oriented tools utilize human domain experts in fact-checking a given news article. These tools curate articles labelled by domain experts. The articles are verified using metrics such as background information, consistency of style and grammar, partisanship, and information veracity (?Politifact.com, 2018). Often, the process of verifying an article requires the consultation of other domain experts who have substantive expertise in the topic at hand to evaluate information that may require specialized knowledge. The source of claim may also be consulted to gain supporting evidence ?. While expert-oriented fact-checking is reliable because of its use of domain experts to verify claims, the reliance on a person to manually verify the article makes the process inefficient and time-consuming. This limits the ability of the tool to scale to an on-demand service that can be used by people on social media (Shu et al., 2017).

Crowdsourcing-oriented tools verify the truthfulness of a claim by using a large volume of volunteers to help annotate content or crowdsourcing. Volunteers can tag sites known to be dubious and questionable as such, and discuss incorrect facts, bad reasoning, or uncivil behavior found in the article (Fiskkit.com, 2018). Some crowdsourcing applications also make use of domain experts to verify the general consensus of volunteers (Fakeblok.com, 2017). While crowdsourced tools are effective in utilizing a wisdom of the crowds approach, it is still inefficient and takes time to arrive at a final verdict. Furthermore, most crowdsourcing-oriented sites and tools still rely on domain experts to do a second check making them scale poorly (Shu et al., 2017). Thus, they are unusable in real time verification

of fake news.

Computational-oriented tools aim to solve the inadequacies that expert and crowdsourcing-oriented tools have. These tools commonly use algorithms and machine learning models to judge a news article as fake in real-time making it scaleable for everyday use. Feng et al. (2012) utilized deep syntax models making use of probabilistic context-free grammars to transform sentences into the underlying rules that the sentences follows. Another method using linguistic analysis is objectivity identification, the centerpiece of the work of Potthast et al. (2017). The work of Jin et al. (2017) builds a credibility propagation network of tweets with supporting or opposing perspectives. More recent approaches also incorporate newer, novel methods to aid in detection. The work of Conforti et al. (2018) handles fake news detection as a specific case of cross-level stance detection. In addition, their work also uses the presence of an inverted pyramid structure as an indicator of real news, using a neural network to encode a given articles structure

While state-of-the-art techniques in fake news detection exist, they still require substantial amounts of resources to make the model perform well. As such, most fake news detection models are trained using English datasets due to the wide availability of the required resources in said language. This creates the need for a localized fake news detector that could identify fake news in languages beyond English. The large resource requirement still exists when localizing these existing methods to other languages.

1.1.2 Low Resource Languages

The work of Cieri et al. (2016) defines low resource languages as languages which lack online and computational resources. In terms of Natural Language Processing, Filipino resources and tools are lacking in number or are non-existent, especially when compared to high resource languages (Cruz & Cheng, 2019). These resources include text datasets, libraries, dictionaries, and any computational research related to the language. Most, if not all modern fake news detection techniques assume the existence of large, expertly-annotated corpora to train models from scratch, which is not present in Filipino. An assessment performed by the group of Cheng (2018) determined that only 25% of the tagged articles in Vera Files are in text format and accessible. Out of the 102 examined text articles, only 7% are in Filipino.

This requirement for large datasets to effectively train fake news detection models from scratch makes it difficult to adapt these techniques into Filipino.

1.1.3 Deep Learning

In recent years, Deep Learning has evolved into a successful field that has made major breakthroughs in multiple areas like Natural Language Processing (Wu et al., 2014), Computer Vision (He et al., 2015), and Speech Recognition (van den Oord et al., 2016). This is due in part to the novel deep neural network architectures that are cornerstones to the discipline. While powerful, these models often necessitate substantial amounts of data that are otherwise not present in some tasks and settings Vinyals et al. (2016).

Few-shot learning is a computer vision technique that aims to learn a model from a small labelled dataset. Formally known as K-shot N-way classification, a classifier is trained to discriminate between N classes when only K samples are provided to make the decision. This is useful in domains where limited data is available (Triantafillou et al., 2017). Often, powerful models suffer from overfitting on small datasets because they were created to accommodate a variety of classification tasks (Vinyals et al., 2016). To avoid overfitting, few-shot learning algorithms use a combination of non-parametric models (such as nearest-neighbors based models) applied on embeddings of the data.

Transfer learning presents another approach to this problem by using language models. Classically, language models tend to overfit to small datasets and suffer from forgetting when finetuned because Natural Language Processing models are generally more shallow than Computer Vision models. ULMFiT (Howard & Ruder, 2018), BERT (Devlin et al., 2018), and GPT-2 (Radford et al., 2019) try to solve these problems by utilizing different finetuning methods and pre-training objectives.

1.2 Research Objectives

1.2.1 General Objective

To localize fake news detection algorithms to Filipino.

1.2.2 Speci c Objectives

1. To collect Filipino text data for a Filipino language model

2. To collect real and fake Filipino news articles for model training and testing
3. To build a baseline Siamese network fake news detection model
4. To build hypotheses ULMFiT, BERT, and GPT-2 fake news detection models
5. To evaluate the fake news detection models

1.3 Scope and Limitations of the Research

The data used to train the language model is Filipino text data sourced from Tagalog wikipedia articles. The wikipedia articles were automatically scraped using a script. All articles are then compiled into one large corpus.

Filipino articles will be used to train the fake news detection models. Articles will be sourced from websites labelled by NUJP as sites commonly disseminating fake news, sites with articles labelled by Vera Files as fake news or misleading, and mainstream media sites as defined by the Philippine government. Most articles were sourced from sites labelled as fake news purveyors instead of being individually expertly-annotated. An equal number of fake and real news articles will be used to train the models. These articles will make up the news corpora.

The baseline fake news detection model utilizes the Siamese network architecture (Koch, 2015). Each Siamese twin was modified to accommodate an additional embedding layer to convert words into vector representations. The model was trained using the collected news corpora.

For comparison, the hypotheses fake news detection models were based on ULMFiT (Howard & Ruder, 2018), BERT (Devlin et al., 2018), and GPT-2 (Radford et al., 2019). The ULMFiT model uses an AWD-LSTM (Merity et al., 2017) trained on a language modeling objective as the base model, BERT uses a masking and sentence pair batching layer, while GPT-2 uses a positional embedding and embedding layer. For BERT and GPT-2, the classification heads is augmented into multitasking heads to compare with the original models. All of the models were trained using the collected news corpora.

To evaluate the models, their testing accuracy will be measured by testing against a hold-out dataset exclusive from the training dataset.

1.4 Signi cance of the Research

This work will make three contributions: First, the first fake news dataset in the low-resourced Filipino language will be constructed to alleviate the data scarcity for research in this domain. Second, this paper shows that transfer learning techniques such as ULMFiT (Howard & Ruder, 2018), BERT (Devlin et al., 2018), and GPT-2 Radford et al. (2019) perform better compared to few-shot techniques and standard RNN-based techniques by good and significantly wide margins, respectively. Third, this paper proposes a novel technique called Stylometric Multitask Transfer which allows transfer learning methods to leverage stylometric information in downstream tasks to finetune more robust and effective classifiers.

Chapter 2

Review of Related Literature

This chapter explores the characteristics, capacities, and restrictions of current research, algorithms, and software similar to this research.

2.1 Existing Fact Checking Tools

This section presents existing fact-checking tools currently being used. Table 2.1 shows the summary of these tools.

The availability of fact-checking tools help in mitigating the spread of fake news (Fakeblok.com, 2017). These tools are usually serviced as websites (Politifact.com, 2018; **?**; Fiskkit.com, 2018) and browser plugins Fakeblok.com (2017).

Some fact-checking tools require experts in their decision making process. An example is Politifacts Truth-O-Meter which utilizes a reporter to analyze facts in certain statements, and an editor that verifies the reporters work. After initial reviews are made, the report is turned over to more editors for further verification. The reporter and editors review the factuality of a statement by verifying the literal truthfulness of a statement. They also examine the possibility to reinterpret the statement, scrutinize given evidence, and reference judgements made for similar statements (Politifact.com, 2018). Fact-checking reports are made public through their website.

Snopes also utilizes experts in fact-checking. To verify any digital content, an editorial staff will undertake preliminary research and background checks on the topic. The written report is passed to copy-editors for style consistency, grammar,

content factuality, and partisanship. Origins of the claim and experts are also contacted to gain more supporting evidence (**?**). Researched claims are posted on their websites.

Fakeblok is another example of a fack-checking tool that consults experts. The web plugin also makes use of crowdsourcing. A user can flag a website as dubious through the plugin. After accumulating several reports, the site will be reviewed by a team of independent journalists. If proven to be fake, the site is flagged by the plugin. Fakeblok will warn users of this site through visual warnings whenever the domains link is visible or the user clicks on the domain link. Fakeblok currently concerns itself with Philippine articles written in either English or Filipino (Fakeblok.com, 2017).

However, not all tools use experts in news verification. Fiskkit is an online platform that facilitates discussion of online articles. The discussions serve as a ground to scrutinize incorrect facts, bad reasoning, and uncivil behavior. Users collaboratively check claims and content to create a granular peer-reviewed discussion on a certain topic. After enough users have discussed and corrected a piece of content, the system generates a page that highlights the flaws in inconsistencies in the text (Fiskkit.com, 2018).

Table 2.1: Existing Fake News Detection Tools

Existing Tool	Type	Evaluation
PolitiFact	Expert-Oriented	Reviews posts and articles by having a reporter fact-check and editors reviewing the fact-check.
Snopes	Expert-Oriented	A series of editors investigate and review a certain article to release a report on its verification.
FakeBlok	Crowdsourced-Oriented and Expert-Oriented	A browser plugin that lets people tag news and sites as fake. An independent board of journalists and experts review the tags to verify fakeness.
Fiskkit	Crowdsourced-Oriented	An online platform where multiple people can simultaneously tag and collaboratively check claims on an article. After enough checks, the system generates a page that highlights the flaws and inconsistencies in the text.

2.2 Fake News Detection Models

This section presents a summary of existing fake news detection models. Table 2.2 shows the summary of these models.

Most deployed fact-checking tools used by the public are not on-demand. Because they utilize experts to verify claims or wait for users to arrive at a consensus, the reports take time to form. Thus, those tools would not be able to immediately label and correct new pieces of content. Computation-oriented tools attempt to remedy this pain by using algorithms and machine learning models.

Numerous recent studies have tackled fake news detection with various techniques. The work of Feng et al. (2012), demonstrated an SVM classification model that detects deception in hotel reviews using (1) Bag-of-words representations including unigrams, bigrams, and the union of the two; (2) POS features with unigram features; (3) Deep syntax features such as unlexicalized production rules, lexicalized production rules, unlexicalized production rules combined with the grandparent node, and lexicalized production rules combined with the grandparent node. The study uses several datasets to explore different types of deceptive writing. The datasets include 400 truthful and deceptive reviews from TripAdvisor-Gold, 400 truthful and deceptive reviews from TripAdvisor-Heuristic, 400 filtered and 400 displayed reviews from Yelp, and 296 essays from Amazon Mechanic Turk. 80% of the resulting dataset is used for training and 20% for testing, with 5-fold cross validation. All features were encoded as TF-IDF values. The sentences were parsed using Berkeley PCFG parser. Results show an increase in performance when utilizing deep syntax features with bag of word representations compared to other baselines that are based on shallow syntax features. The model reached 91.2% accuracy on hotel review data.

The work of Bourgonje et al. (2017) identifies and verifies the stance of a headline with respect to its content. They make use of the first Fake News Challenge dataset ("Ferreira:", n.d.-b) consisting of 1,668 articles and 1,648 unique headlines resulting into 49,962 annotated pairs. First, they determined whether the headline and article is related using n-gram matching of the lemmatised input using the CoreNLP Lemmatiser (Manning et al., 2014), with a 90% training and 10% training set division, validated with 50-fold cross-validation. They achieved an accuracy of 89.59% using relatedness scores.

Karimi et al. (2018) uses a deep learning approach and integrates multiple sources to assign a degree of fakeness to an article. They use a CNN to extract local patterns from a text then apply an LSTM on top of the extracted features to capture temporal dependencies. They then use an interpretable multi-source

fusion to combine features from different sources. Afterwards, they cluster similar samples together and separate dissimilar. The work utilizes the LIAR dataset (Wang, 2017). The proposed work outperforms baselines and is effective in integrating information from multiple sources.

Unlike the previous studies which based their work on text features alone, Jin et al. (2016) constructed a network to verify an articles credibility. The network is a credibility propagation network of tweets from Twitter as the nodes. Supporting edges are created between nodes when they take the same viewpoint. Opposing edges are created when nodes take different viewpoints. The network is composed of 49,713 tweets from 42,310 unique users. Tweets were modelled as topic-viewpoint pairs with each pair represented by a probability distribution over document terms. Pairs are then clustered into conflicting viewpoint clusters. Afterwards, distance between pairs with the same topic are computed as the Jensen-Shannon Distance to be compared with a predefined threshold to form cannot-link restraints. As a result, similar viewpoints are clustered together and conflicting pairs are separated. The network was evaluated using (1) Classification accuracy of articles, (2) Precision and recall for fake and real news, respectively, and (3) F1 score. The papers model performs very consistently despite topic variations but is outperformed by other baseline models.

To remedy the inability of network features to stand alone in fake news detection, Ruchansky et al. (2017) utilized (1) Text features, (2) Temporal features, and (3) User features when creating their CSI model. Text features are articles represented as doc2vec vectors. Temporal features include the number of engagements and time between engagements which were given to an RNN to capture temporal patterns. User features include scores derived from a weighted user graph with edges denoting the number of articles two users have both engaged. Their work made use of posts from Sina Weibo and tweets from Twitter. The collected data amounted to 5,656 articles with 3,053,057 users participating in the discussion. Results indicated that the combination of all 3 modules (Capture, Score, Integrate) performed the best over other variations. The proposed model also showed better performance when compared to other baseline models. Results were validated using 5-fold cross validation.

Several other studies also tackled fake news detection by incorporating novel techniques to aid in detection. Potthast et al. (2017) developed a model that considered writing style and political orientation when detecting fake news. Their work explores writing style features such as n-grams, stop words, and POS tags. Furthermore, they employ readability scores and dictionary features indicating word frequency. Domain-specific features including quoted word and external link ratios, number of paragraphs, and average paragraph length were also included. 1,627 articles were taken from Buzzfeed-Webis Fake News Corpus as the dataset.

To prevent bias, they balanced the training sets using oversampling. Furthermore, they perform 3-fold cross-validation where each fold contains a publisher from each political orientation. Results show that all classifiers outperforms naive baselines but at the cost of a large decrease in recall. Orientation-specific style and topic classifiers perform better than general classifiers. But none of them outperform naive baselines in F-measure. The paper concluded that style-based fake news classification does not work in general.

The work of **?** handles fake news detection as a specific case of cross-level stance detection. Each article is converted into its embedding representation then encoded using BiLSTM. The relationship between the headline and article is then evaluated by first using double-conditional encoding, connected using attention, and soft-selected the relevant elements using self-attention. The sentence vector representations are then aggregated using a backward LSTM. The data was taken from the FNC-1 corpus with 49,972 labelled stances for each headline-body pair. Articles are split into sentences using NLTK sentence tokenizer. The study uses precision, recall, and F1 score to measure model performance. The best performance was obtained using the co-matching attention model using only word embeddings.

While these approaches are valid and robust, most, if not all modern fake news detection techniques assume the existence of large, expertly-annotated corpora to train models from scratch. Both Bourgonje et al. (2017) and Conforti et al. (2018) used the Fake News Challenge dataset, with 49,972 labelled stances for each headline-body pair. Karimi et al. (2018) used the LIAR dataset (Wang, 2017), which contains 12,836 labeled short statements as well as sources to support the labels, while Jin et al. (2016) used 49,713 tweets supported by 42,310 unique users. Ruchansky et al. (2017) used 5,656 articles with 3,053,057 users to train his model. Potthast et al. (2017) used the Buzzfeed-Webis Fake News Corpus 2016, containing 1,627 labelled articles. Feng et al. (2012) used 2,400 reviews supplemented by 296 essays.

Table 2.2: Related Works on Fake News Detection Models.

Related Work	Goal	Features	Dataset	Evaluation
Feng et al. (2012)	Detect deception via bag of words and syntactic stylometry	Unigrams, Bigrams, Part of Speech Tags, Deep Syntax Rules, (Un)lexicalized Production Rules	TripAdvisor Gold reviews, TripAdvisor Heuristic reviews, Yelp reviews, and Essays	5-fold cross validation. Results show an increase in performance when adding deep syntactic features
Bourgonje et al. (2017)	To identify and verify the stance of a headline with respect to its content	Relatedness score, Three-class score, Weighted score	Fake News Challenge corpus	50-folds cross validation. Using Logistic Regression followed by 3 binary classifiers gave the best result.
Karimi et al. (2018)	To integrate multiple sources when assigning a degree of fakeness to an article	Automatic feature extraction	LIAR dataset	Accuracy and f-measure. Their proposed model performed the best compared to other traditional models.
Jin et al. (2016)	To verify a news articles credibility using propagation networks	Credibility network, Type of link (Supporting or opposing), Link degree	Sina Weibo dataset	Accuracy, Precision, Recall. The papers model performs very consistently despite topic variations but is outperformed by other models.

Ruchansky et al. (2017)	Classify fake news and recognize groups of malicious users	Textual and corpus features, Temporal features (interactions) and User features (incidence matrices)	Twitter news articles, Twitter tweets, Weibo discussion topics	5-folds cross validation. The combination of all 3 modules showed the best performance.
Potthast et al. (2017)	Assess style similarity between two types of partisan news and Implement a style based fake news detector	N-grams, Stop words, Ratios of of external links and quoted words, Number of Paragraphs, Average length of paragraph in document	Buzzfeed-Webis Fake News Corpus	3-fold validation. Style and topic based classifiers fared better than naive classifiers
Conforti et al. (2018)	Uses cross-level stance detection to detect fake news	Named entities, Characters	Fake News Challenge corpus	Precision, Recall, and F1 score. Best performance was co-matching attention model using only word embeddings

2.3 One-shot and Few-shot Learning

This section presents a detailed discussion on One-shot and Few-shot models, the context where they were first proposed, and their applications in deep learning fields. Table 2.3 shows the summary of these models.

2.3.1 One-Shot Learning

Several approaches exist when tackling a classification task where only a limited number of samples are given in order to inform a decision by a model.

Koch (2015) used Siamese Neural Networks to perform one-shot classification on images. Siamese Networks consists of twin neural networks that compute an embedding of an image, which are connected by an energy-based, contrastive loss function at the top that measures similarity of image pairs fed into the twin networks.

The work of Vinyals et al. (2016) suggested Matching Networks to execute one-shot classification on images by using an attention mechanism to enable rapid learning of novel classes from a support set of examples. Unlike the pair-wise training method of Koch (2015), Matching Networks could learn to classify novel classes from a single example by computing embeddings of a sample into a vector space.

The work of Snell et al. (2017) extended the task to incorporate more samples, into what is called few-shot learning, or more formally as K-shot N-way learning. To classify N classes with only K samples provided, they proposed Prototypical Networks. By computing "prototypes" of a class derived from the mean vector of all embedded representations of each sample of a class in a vector space, a nearest-neighbors based approach can be used for a new query sample. Prototypical Networks attain the same state-of-the-art results of Matching Networks without using computationally expensive methods and architectures.

Finn et al. (2017) took a different approach to few-shot learning. The proposed a model-agnostic meta-learning model that learns a network initialization that can quickly adapt to new tasks. Instead of learning on batches of samples, MAML learns on batches of tasks. However, this technique is memory and compute intensive.

2.3.2 Natural Language Processing Applications

Most K-shot N-way learning work are within the field of Computer Vision, where the method was first pioneered in 2003 by Fe-Fei et al. (2003), taking a bayesian approach for one-shot learning of image categories. By taking advantage of a few examples, the approach does not necessitate the large datasets required by most deep learning architectures and models.

The work of Yu et al. (2018) used adaptive metric-learning techniques to automatically determine the best weighted combination from a set of metrics obtained from a number of meta-training tasks. A model is trained for multiple "meta-training tasks" where each task is a few-shot learning task that aims to classify sentiments from an online reviews dataset.

The work of Bailey & Chopra (2018) adapts the concept of prototypes proposed by Snell et al. (2017), but instead of prototypes fully computed automatically, a human is introduced in order to influence and refine the prototype choosing.

Table 2.3: Related Works using One-Shot and Few-Shot Learning.

Related Work	K-Shot Approach	Domain	Usage
Koch et al. (2015)	One-Shot Classification via Siamese Networks.	Computer Vision	Learn similarities between pairs of images to classify novel classes.
Vinyals, et al. (2016)	One-Shot Classification via Matching Networks	Computer Vision	Learn embeddings of classes via an attention-based mechanism.
Snell, et al. (2017)	Few-Shot Classification via Prototypical Networks	Computer Vision	Compute prototypes which can be queried using nearest-neighbors.
Finn, et al. (2017)	Few-Shot Classification via Model-Agnostic Meta-Learning	Computer Vision	Learns from batches of tasks.
Yu, et al. (2018)	Few-Shot Classification via Adaptive Metric Learning Techniques	NLP	Use Adaptive Learning Techniques on Matching Networks to classify text.
Bailey, et al. (2018)	Few-Shot Classification via Prototypical Networks and a Human-in-the-Loop	NLP	Use Prototypical Networks refined by a human that influences the prototype creation.

2.4 Transfer Learning

This section presents a detailed discussion on transfer learning models, the context where they were first proposed, and their applications in deep learning fields. Table 2.4 shows the summary of these models.

ULMFiT (Howard & Ruder, 2018) was introduced as a transfer learning method for Natural Language Processing that works akin to ImageNet (Russakovsky et al., 2014) pretraining in Computer Vision. It uses an AWD-LSTM (Merity et al., 2017) pretrained on a language modeling objective as a base model, which is then finetuned to a downstream task in two steps. First, the language model is finetuned to the text of the target task to adapt to it syntactically. Second, a classification layer is appended to the model and is finetuned to the classification task conservatively. During finetuning, multiple different techniques are introduced to prevent catastrophic forgetting, wherein the model loses most (if not all) information and relations it has learned during the pretraining stage. ULMFiT holds state-of-the-art for text classification, and is notable for being able to set comparable scores with as little as 1000 samples of data which makes it attractive for use in low-resource settings.

BERT (Devlin et al., 2018) is a transformer-based (Vaswani et al., 2017) language model that is designed to pretrain deep bidirectional representations that can be finetuned to different tasks, with state-of-the-artresults achieved in multiple benchmarks (Devlin et al., 2018). BERTs power comes from Attention, a mechanism that allows a network to give more weight to certain tokens in a sequence, essentially paying more attention to important parts (Vaswani et al., 2017). Attention allows BERT to model not only sequences, but also the importance and weight of each token in a sequence with respect to other sequences, as well as itself. In addition to leveraging Attention, it uses the Transformer (Vaswani et al., 2017) architecture, to where BERT gains its bidirectionality. Transformers are sequence models that do not use recurrent layers, instead leveraging only feed-forward layers and attention mechanisms. The disuse of recurrences provide two advantages: First, it allows transformers to be parallelized as they are not sequential in nature unlike LSTMs or GRUs. Second, they allow batches of text to be seen at once, again due to its unsequential nature, which also in turn allows it to leverage attention mechanisms and be bidirectional. BERT is unique that it uses modified tasks for pretraining. Given that its bidirectionality gives it access to left-context, the model would be able to peek directly at the next words when following a standard language modeling task. To alleviate this, the authors propose the use of masked language modeling, which masks a number of words in the sentence with the model tasked to identify them (Devlin et al., 2018). In addition, a second pretraining task called next-sentence prediction was added to

enforce stronger relationships between two sentences. In this task, a target sentence is identified if it is likely to proceed a source sentence (Devlin et al., 2018). In addition to these augmentations, BERT also benefits from being deep, allowing it to capture more context and information.

The GPT-2 (Radford et al., 2019) architecture builds up from the original GPT (Radford et al., 2018) model. Its main contribution is the way it is trained: with an improved architecture, it learns to do multiple tasks by just training on language modeling. Architecture-wise, it is a transformer-based model similar to BERT, with a few differences. It uses two feed-forward layers per transformer block, in addition to utilizing layer normalization (Ba et al., 2016) on the input matrices of each block akin to a residual layer (He et al., 2015). In terms of pretraining tasks, it only uses plain language modeling unlike BERT, which uses masked-language modeling and next sentence prediction. GPT-2 is notable for being extremely deep, with 1.5B parameters, 10x more than the original GPT architecture. This gives it more flexibility in learning tasks unsupervised from language modeling, especially when trained on a very large unlabeled corpus.

Table 2.4: Related Works using Transfer Learning.

Related Work	Model or Technique	Contribution
Howard et al. (2018)	ULMFiT	Proposed discriminative finetuning, slanted triagnular learning rates, and gradual unfreezing to avoid catastrophic forgetting during finetuning.
Devlin et al. (2018)	BERT	Proposed a bidirectional language model using masked language modeling.
Radford et al. (2019)	GPT-2	Proposed a unidirectional language model that can learn multiple tasks by just training on a general language corpus and finetuned to the task corpus.

2.5 Towards a Localized Fake News Detector

While state-of-the-art models in fake news detection exist, most of their methodologies assume the existence of a large volume of available resources, particularly in text. This would make localization to a low-resource language hard, as data is expected to be scarce.

To circumvent the problem of resource scarcity, the proponents propose the use of transfer learning to train a model that can generalize from a small amount of data. To add, the proponents will also demonstrate the capability of few-shot learning as a good baseline in low-resourced fake news detection.

Chapter 3

Theoretical Framework

This chapter contains details on the theory and implementation of algorithms and techniques that are critical within the research.

3.1 Word Embeddings

In order to vectorize words and phrases in the study, the Word2Vec model Mikolov et al. (2013), specifically its variant GloVe Pennington et al. (2015) will be used.

3.1.1 Overview

For words to be used in various machine learning tasks, it must first be converted into a corresponding representation, usually where each word is regarded as an atomic unit, such as vocabulary indices. This poses a number of disadvantages, notably the lack of the notion of word-similarity and relationship in the learned model.

The current state-of-the-art involves the use of Word2Vec Mikolov et al. (2013) and its variants. This technique involves learning a representation of words in a vector space where each related word appears closer to words that are related to it, learned via a neural network.

The network comprises of four layers: an input layer, a projection layer, a hidden layer, and an output layer. Given input words, the neural network will project the corpus in a vector space, further refine it, then output a continuous

vector representation of the corpus.

The work of Pennington et al. (2015) builds upon this by using a specific weighted least-squares model trained on global word-to-word co-occurence counts from the corpus, instead of pure word-to-word pairs. The model is named "GloVe," coined from "Global Vectors," as the model captures global corpus statistics.

3.1.2 Notation

Notation is established by Pennington et al. (2015). Given a matrix of word-to-word co-occurence counts X where $X_{i;j}$ denotes the number of times the word j appears within the context of the word i, $X_i = \sum_k X_{i;k}$ indicates the number of times any word occurs within the context of word i. $P_{i;j} = P(j|i) = \frac{X_{i;j}}{X_i}$ is the probability that word j appears within the context of the word i.

It is argued that the suitable starting point for learning word vectors would be the ratios of co-occurence probabilities instead of word probabilities Pennington et al. (2015).

Given three words i, j, and k (two words we are trying to place in the vector space and a third reference or query word), the general model would take the form:

$$F(w_i; w_j; \tilde{w}_k) = \frac{P_{i;k}}{P_{j;k}}$$

where $w \in \mathbb{R}^d$ denote word vectors and $\tilde{w} \in \mathbb{R}^d$ denote context word vectors. The right hand side of the equation is derived from the corpus and F is a function whose parameters are currently unspecified, but should encode the information present within the ratio $P_{i;k}/P_{j;k}$ in the embedding space.

3.1.3 Derivation of the Word Vector Model

To find a function F that encodes the information present in the ratio $P_{i;k}/P_{j;k}$, we further refine the general model presented in the previous subsection Pennington et al. (2015).

Considering that vector spaces are linear structures, we can modify the model to restrict the possible functions F into functions that rely on the difference of

two target words in the corpus. This modifies the model into:

$$F(w_i \quad w_j; \tilde{w}_k) = \frac{P_{i;k}}{P_{j;k}}$$

Since the left hand side of the equation is a vector and the right hand side is a scalar, the function could obfuscate the linear structure the model is attempting to capture. In order to prevent the function from mixing the vector dimensions, we then take the dot product of the arguments:

$$F((w_i \quad w_j)^T \tilde{w}_k) = \frac{P_{i;k}}{P_{j;k}}$$

Do note that word-to-word co-occurence matrices have arbitrary distinctions between a context word and a word, meaning they are interchangeable. To maintain consistency, both $w \ \$ \ \tilde{w}$ and $X \ \$ \ X^T$ should be interchangeable. If such relabelling would be applied to the current model, it would render it invariant, thus symmetry must be restored by requiring F be a homomorphism between the groups $(\mathbb{R}; +)$ and $(\mathbb{R}_{<0}; \ )$, i.e.:

$$F((w_i \quad w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

Which using our current model equation, is solved by:

$$F(w_i^T \tilde{w}_k) = P_{i;k} = \frac{X_{i;k}}{X_i}$$

Solving the actual equation gives us the following model:

$$w_i^T \tilde{w}_k = log(P_{i;k}) = log(X_{i;k}) \quad log(X_i)$$

The $log(X_i$ term on the right hand side prevents the exhibition of the exchange symmetry. However, the term can be absorbed into a bias term b_i for w_i because it is independent from k. To restore the symmetry, an additional bias term is added:

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = log(X_{i;k})$$

While this equation simplifies the original model for use in learning, it encounters a number of problems. First, the logarithm diverges whenever its argument is zero. To solve this, an additive shift may be included into the logarithm $log(X_{i;k} \;\; ! \;\; log(1 + X_{i;k}$ as sparsity of X is maintained and logarithmic divergence is avoided. However, this model will encounter the drawback of weighing all co-occurences equally, which could lead to noisy vectors that could carry less information.

The main contribution of Pennington et al. (2015) is proposing a weighted-squares regression model that addresses the aforementioned problems. Turning the current model into a least-squares problem and including a weighting function $f(X_{i;j})$ in the loss function results into the final model:

$$J = \sum_{i;j=1}^{V} f(X_{i;j})(w_i^T \tilde{w}_j + b_j + \tilde{b}_j \quad log(X_{i;j}))^2$$

where V is the vocabulary size.

3.1.4 Training

The work of Pennington et al. (2015) prescribes the following mechanics for the training of a GloVe model.

Loss Function. The loss function is derived in the previous subsection, and is detailed as:

$$J = \sum_{i;j=1}^{V} f(X_{i;j})(w_i^T \tilde{w}_j + b_j + \tilde{b}_j \quad log(X_{i;j}))^2$$

Given the function F in the original model as a parametric-function, a neural network is trained to optimize the parameters using a corpus of words of size V subject to the aforementioned loss function.

Optimization. The network is trained using AdaGrad Duchi et al. (2011) while stochastically sampling non-zero elements from X. For vectors of less than 300 dimensions, 50 iterations are done. Otherwise, 100 iterations of training are done. A context vector of 10 words to the left and 10 words to the right are to be used.

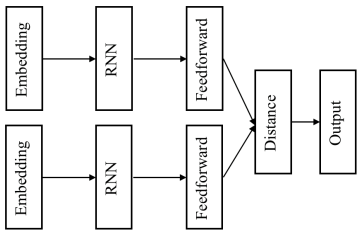

Figure 3.1: Siamese Neural Network Architecture. A pair of twin neural networks learn representations of input pairs and is compared using a loss function connecting both networks. Adapted from Koch (2015)

The model produces two word vectors W and $\tilde{W}$ that are equivalent when X is symmetric. The final word vector output would be taken by summing $W + \tilde{W}$. This output vector can now be used for numerous Natural Language Processing tasks Pennington et al. (2015).

3.2 Siamese Networks

Siamese Networks for One-Shot Learning Koch (2015) will be used as the baseline model for the study.

3.2.1 Overview

Siamese Neural Networks were originally conceived to solve the signature verification problem by viewing it as an image matching problem Bromley et al. (1993), and was subsequently adapted as a one-shot learning technique for use in Computer Vision Koch (2015).

To train a Siamese Network for performing one-shot classification, it first has to be trained to discriminate class-identity between pairs of images. Networks that perform well in this similarity task should be able to generalize well to perform one-shot classification Koch (2015). To evaluate new images, it is tested for similarity against all other learned images in the test class.

3.2.2 Model Architecture

A Siamese Neural Network is a pair of twin neural networks that learn representations of a pair of inputs, and compares them using a loss function that connects the final layers of both networks. This loss function is usually an energy-based, contrastive loss function that discriminates between the similarity of the learned embeddings of the pair of inputs Koch (2015).

The study of Koch (2015) utilizes deep convolutional neural networks as a basis for each Siamese twin. Each twin comprises of a series of convolutional blocks. Each block comprises of a convolutional layer with a fixed stride of 1 and filters of various sizes. A ReLU activation function is then applied to the output feature mappings, which is succeeded by a max-pooling with a stride and filter size of 2.

The nth filter map in each layer results into the form:

$$a^{(k)}_{1;m} = \text{max-pool}(max(0; W^k_{l\ 1;l} \ ? \ h_{1;(l\ \ 1)} + b_l); 2)$$

$$a^{(k)}_{2;m} = \text{max-pool}(max(0; W^k_{l\ 1;l} \ ? \ h_{2;(l\ \ 1)} + b_l); 2)$$

Where $W_{l\ 1;l}$ indicates a 3-dimensional tensor that represents the mapped features for layer l, $h_{1;l}$ indicates the hidden vector in layer l of the first Siamese twin, $h_{2;l}$ indicates the hidden vector in layer l of the second Siamese twin, and $?$ is a valid convolution operation that corresponds to returning those output units that resulted from the complete overlap between the input feature maps and each convolutional filter. Each Siamese twin is a network with L layers each with N_l units.

The units in the final layer of the last convolutional block are flattened into a single vector, and succeeded by a layer which is fully-connected. There is an additional layer that receives the input of both Siamese twins, which computes a distance metric between the learned embeddings of both input pairs. This is then presented to a single sigmoid output unit.

The prediction vector is now given as:

$$p = \ (\sum_j \ _j j h^{(j)}_{1;L\ \ 1} \quad h^{(j)}_{2;L\ \ 1} j)$$

Where denotes the sigmoidal activation function.

The architecture can be seen in figure 3.1

3.2.3 Training

The work of Koch (2015) prescribes the following mechanics for the training of Siamese Networks.

Loss function. Given a minibatch of size M where i indexes the ith minibatch, $y(x_1^{(i)}; x_2^i)$ represents a length-M vector containing the minibatch labels. For this minibatch, assume that $y(x_1^{(i)}; x_2^i) = 1$ if both input pairs are from the same class, and $y(x_1^{(i)}; x_2^{(i)}) = 0$ if the input pairs are from different classes. A regularized cross-entropy loss function is then imposed on the network in the following form:

$$L(x_1^{(i)}; x_2^{(i)}) = y(x_1^{(i)}; x_2^{(i)}) \log p(x_1^{(i)}; x_2^{(i)}) + (1 \quad y(x_1^{(i)}; x_2^{(i)})) \log (1 \quad p(x_1^{(i)}; x_2^{(i)})) + \quad ^{\intercal} jwj^2$$

Optimization. The network is trained using a standard stochastic gradient-descent based algorithm, in this case, Adam Kingma & Ba (2014) with a minibatch size of 128, a learning rate $\ _j$, a momentum of $\ _j$, and with L_2 regularization weights $\ _j$ layer-wise.

The theta-update rule is then defined per epoch T:

$$\begin{aligned} {}^{(T)}_{kj}(x_1^{(i)}; x_2^{(i)}) &= \ {}^{(T)}_{kj} + \Delta \ {}^{(T)}_{kj}(x_1^{(i)}; x_2^{(i)}) + 2 \ _j j \ _{kj} j \\ \Delta \ {}^{(T)}_{kj}(x_1^{(i)}; x_2^{(i)}) &= \quad _j \frac{@}{@ \ _{kj}} \ {}^{(T)}_{kj} + \ _j \Delta \ {}^{(T \ 1)}_{kj} \end{aligned}$$

3.3 ULMFiT

ULMFiT will be one of the hypothesis models used for the study.

3.3.1 Overview

Because transfer learning was proposed in Computer Vision, models adapted to Natural Language Processing tend to overfit and lose knowledge during finetuning. ULMFiT was proposed as an effective transfer learning method for Natural Language Processing (Howard & Ruder, 2018). They use novel techniques to

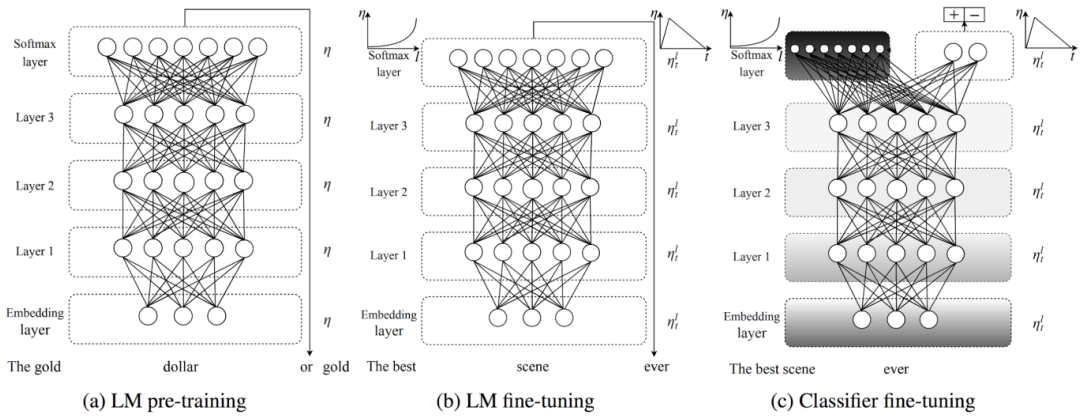

Figure 3.2: Overall ULMFiT pretraining and finetuning framework. An AWD-LSTM (Merity et al., 2017) is pretrained on a language modeling task. The weights are then reused with no modifications to the architecture. For finetuning, the model is first finetuned, again using language modeling, this time to the text of the target dataset to adapt to its own vocabulary and idiosyncracies. Lastly, a classification layer is added to the model and is finetuned for text classification. Adapted from Howard & Ruder (2018)

address catastrophic forgetting whilst retaining the same architecture ImageNet models have.

3.3.2 Model Architecture

The model is a 3-layer AWD-LSTM language model (Merity et al., 2017) with an embedding layer appended at the top. Each layer of the model is a weight-drop LSTM which is an LSTM that uses DropConnect (Wan et al., 2013) on the recurrent hidden-to-hidden weight matrices.

The architecture is the same during pretraining and finetuning. The architecture can be seen in figure 3.2

3.3.3 Pretraining

Because the architecture was based on ImageNet, the training corpus should be large, ImageNet-like, capturing general properties of the language. As such, wikipedia articles work well because they are general articles. For this work, the collected TagalogWikipedia corpora is used to pretrain ULMFiT.

3.3.4 Finetuning

Finetuning will be divided into two downstream tasks: target task finetuning and target task classifier finetuning. Techniques such as slanted triangular learning rates, concat pooling, gradual unfreezing, and backpropagation through time will be used during those tasks. Both left-to-right and right-to-left language models will be pretrained. Their predictions will be averaged.

Target Task Finetuning

Despite the diversity of the general-domain data used for pretraining, the target task data will likely come from a different distribution. Thus, the language model should be finetuned on the target task data to adapt it to its indiosyncrasies. This results into a robust language model even for small datasets.

Discriminative Finetuning. Different layers should be finetuned to different extents because they capture different types of information. Discriminative finetuning allows different layers to tuned with different learning rates.

A regular stochastic gradient descent update looks like the following with at time step t:

$$_t = _{t\ 1} \quad r_{\ (\ )}$$

with as the learning rate and $r_{\ (\ )}$ as the gradient with respect to the model's objective function. Discriminative finetuning splits the parameter into $^1; \ldots; ^L$ where ^{L} contains the model's parameters at the l-th layer with L as the number of layers. Similarly, is split to $^1; \ldots; ^L$ where is the learning rate of the l-th layer. The equation is then updated to the following:

$$^l_t = ^l_{t\ 1} \quad ^l \quad r_{\ l}J(\ )$$

Howard & Ruder (2018) recommends finetuning only the last layer, and using $^{l\ 1} = ^l/2.6$ as the learning rate for the lower layers and ^{L} for the last layer

Slanted Triangular Learning Rates. Using the same learning rate throughout training is not ideal when making the model quickly converge. Thus, (Howard & Ruder, 2018) modifies triangular learning rates as proposed by Smith (2015) with a short increase and a long decay period. The modified triangular learning

rates called slanted triangular learning rates is as follows:

$$cut = bT \quad cut\_frac$$

$$f(x) = \begin{cases} t=cut; & \text{if t} < \text{cut} \\ 1 \quad \dfrac{t \quad cut}{cut \quad (1=cut\_frac \quad 1)}; & \text{otherwise} \end{cases}$$

$$= \frac{1+p \quad (ratio \quad 1)}{ratio}$$

where cut is the iteration when switching from increasing to decreasing the learning rate, T is the number of training iterations, cut_frac is the fraction of iterations where learning rate was increased, p is the fraction of iterations where learning rate was increased or will decrease respectively, is the learning rate at iteration t, is the maximum learning rate, and $ratio$ indicates the difference between the lowest learning rate from the highest.

Howard & Ruder (2018) recommends to use $cut\_frac = 0.1$, $ratio = 32$ and $= 0.01$.

Target Task Classi er Finetuning

. The classifier is finetuned by augmenting the pretrained language model with two additional linear blocks. Each block uses batch normalization and dropout, with a ReLU activation for layers in between and a softmax activation that outputs a probably distribution. The pooled last layer states is taken as input by the first linear layer.

Concat Pooling. Because signals in text classification tasks can occur anywhere in the document, information may get lost if only the last hidden state of the model is considered. Thus, the hidden state at the last time step h_T of the document with both the max-pooled and the mean-pooled representation of the hidden states over as many time steps as possible for the GPU memory $H = h_1; :::; h_T$:

$$h_c = [h_t; maxpool(H); meanpool(H)]$$

where [] is concatenation.

Gradual Unfreezing. To avoid catastrophic forgetting, the model is gradually unfrozen starting from the last layer and all unfrozen layers for one epoch.

This process is repeated until all layers are unfrozen and finetuned until convergence for the last iteration.

Backpropagation through Time. To make classifier finetuning feasible for large documents, each document is divided into fixed-length batches of size b. When starting each batch, the model is initialized with the final state of the previous batch. Meaning hidden states for mean and max-pooling are kept track of and gradients are back propagated to batches whose hidden states contributed to the final prediction.

3.4 BERT

BERT will be one of the hypothesis models used for the study.

3.4.1 Overview

To pretrain word embeddings vectors, left-to-right or right-to-left language modeling objectives have been used as well as discriminating words in left and right context. When using only a left-to-right or right-to-left approach, words lose contextual information. However, using both is inefficient because of pretraining costs and the resulting concatenation is not deep.

BERT uses masked language models for pretraining, enabling deep bidirectional representations (Devlin et al., 2018).

3.4.2 Model Architecture

BERT is a multilayer bidirectional Transformer encoder based on the original Transformer architecture of Vaswani et al. (2017).

The tranformer uses embeddings to convert input and output tokens to vectors. Because there is no recurrences or convolution, the transformer uses positional encoding to inject information about the relative or absolute position in the sequence. The positional encoding can be summed with the embeddings because they have the same dimension. The resulting vectors are passed on to transformer blocks.

Each transformer block is comprised of 2 layers. The first is a multihead

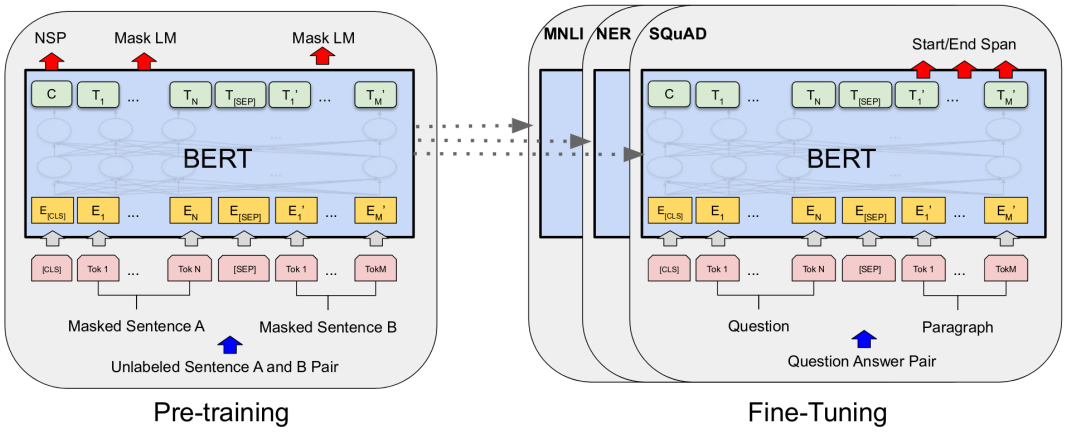

Figure 3.3: Overall BERT pretraining and finetuning framework. Note that the same architecture in pretraining is also used in finetuning with little-to-no modification in structure. After masked-language model and next-sentence prediction is pretrained, we transfer the weights of the model to downstream tasks, with question answering and entailment shown in this example. Adapted from Devlin et al. (2018)

attention mechanism which allows the model to look at information from different representation spaces at different positions. Multihead attention is represented as:

$$MultiHead(Q;K;V) = Concat(head_1;:::;head_h)W^O$$
$$wherehead_i = Attention(QW_i^Q;KW_i^K;VW_i^V)$$

where Q is the queries, K is the keys, and V is the values if the dimension.

The second layer is a position wise fully connected feed-forward network. The network consists of two linear transformations with a ReLU activation in between. The feed-forward network is as follows:

$$FFN(x) = max(0;xW_1 + b_1)W_2 + b_2$$

A residual connection and layer normalization is also employed around each of the layers.

BERT architecture is depicted in Figure 3.3

3.4.3 Pretraining

Pretraining BERT requires the use of two unsupervised tasks namely masked language modeling and next sentence prediction. BERT will be pretrained on the TagalogWikipedia corpora.

Masked Language Modeling. Classically, standard language models can not be trained bidirectionally since it would allow each word to see itself in a multilayer context. Thus, the model could trivially predict the target word. To train a bidirectional representation that does not involve concatenating both a left-to-right and right-to-left language models, some percentage of the input tokens are masked at random. Those masked tokens are then predicted. For BERT, the final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary. Only the masked words are predicted instead of reconstructing the entire input.

Although masking allows bidirectionality for the language model, it creates a mismatch between pre-training and finetuning. To alleviate this, not all masked words are replaced with $[MASK]$ tokens. Only 15% of token positions at random are chosen for prediction. Of those chosen, only 80% are replaced with the $[MASK]$ token, 10% are replaced with a random token, and 10% are unchanged. T_i will be used to predict original token with cross entropy loss

Next Sentence Prediction. BERT is pretrained for a binary next sentence prediction task to enable understanding of sentence relationships. Given a pair of sentences A and B, BERT predicts if sentence B is the actual next sentence of sentence A.

3.4.4 Finetuning

Finetuning is simply swapping out the appropriate input-output pairs for the task. The pairs are encoded using self-attention mechanisms effectively including bidirectional cross attention between the two sentences.

For each task, task-specific input and output pairs are plugged into BERT and all the parameters are finetuned end-to-end. For this study, an input-output pair is an article body and label pair.

3.5 GPT-2

GPT-2 will be one of the hypothesis models used for the study.

3.5.1 Overview

Most models implement task conditioning in their architectures which makes it difficult to tweak for different tasks. But for language, tasks, inputs, and outputs can be represented as a sequence of symbols.

In principle, language modeling is able to perform many different tasks using the format $p(output|input; task)$ without the need for explicit supervision. However, the problem lies in optimizing the unsupervised objective to convergence. Preliminary experiments of unsupervised approaches show that multitask learning is possible given that it is sufficiently large. However, it is much slower than in explicitly supervised approaches. GPT-2 demonstrates that a language model trained on a WebText dataset can perform different tasks demonstrated in natural language sequences, regardless of procurement method.

3.5.2 Model Architecture

GPT-2 uses the same Transformer (Vaswani et al., 2017) architecture as BERT with the exception of information flow. BERT uses a bidirectional transformer which enables it to learn both left and right token contexts while GPT-2 uses a left-to-right transformer.

The tranformer uses embeddings to convert input and output tokens to vectors. Because there is no recurrences or convolution, the transformer uses positional encoding to inject information about the relative or absolute position in the sequence. The positional encoding can be summed with the embeddings because they have the same dimension. The resulting vectors are passed on to transformer blocks.

Each transformer block is comprised of 2 layers. The first is a multihead attention mechanism which allows the model to look at information from different representation spaces at different positions. Multihead attention is represented as:

$$MultiHead(Q; K; V) = Concat(head_1; :::; head_h)W^O$$

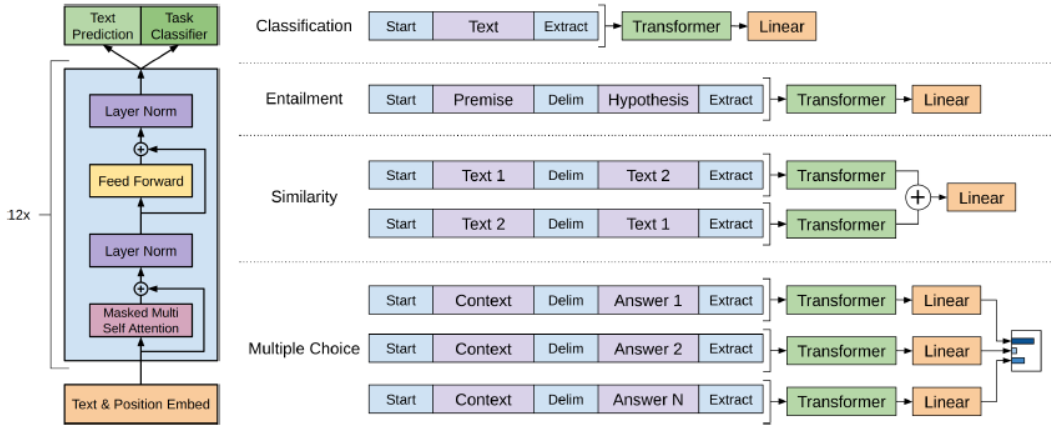

Figure 3.4: Overall GPT/GPT-2 pretraining and finetuning network. We can see a GPT-Transformer with 12 blocks being pretrained on text prediction (language modeling). Different pipelines are then used to finetune them to four different tasks, including classification, textual entailment, textual similarity, and question answering. In this paper, use the classification pipeline to finetune a GPT-2 transformer for fake news detection. Adapted from Radford et al. (2019)

$$wherehead_i = Attention(QW_i^Q; KW_i^K; VW_i^V)$$

where Q is the queries, K is the keys, and V is the values if the dimension.

The second layer is a position wise fully connected feed-forward network. The network consists of two linear transformations with a ReLU activation in between. The feed-forward network is as follows:

$$FFN(x) = max(0; xW_1 + b_1)W_2 + b_2$$

A residual connection and layer normalization is also employed around each of the layers.

GPT-2 architecture is depicted in Figure 3.4

3.5.3 Pretraining

GPT-2 is pretrained on the TagalogWikipedia corpora with language modeling as its sole task.

Language Modeling. Given a set of sentences $(x_1; x_2; \ldots; x_n)$, each with a variable length sequences of $(s_1; s_2; \ldots; s_n)$, language can be represented as the product of conditional probabilities

$$p(x) = \prod_{i=1}^{\gamma} p(s_n | s_1; \ldots; s_{n-1})$$

since language has a natural sequential ordering.

3.5.4 Finetuning

To train the model on fake news detection, GPT-2 is finetuned on the Fake News corpora using the pretrained weights with an appended classification head.

3.6 Stylometric Multitask Finetuning

An advantage of ULMFiT over transformer-based techniques like BERT and GPT-2 is its inherent language model finetuning step. In this step, the model is said to adapt to the idiosyncracies of the target task data (Howard & Ruder, 2018), which in turn allows it to perform better.

In the next chapter, the proponents provide evidence for this assumption with an experiment, and propose a way to add a language model finetuning step to transformer-based models without sacrificing extra finetuning and compute time.

Taking inspiration from multitask learning (McCann et al., 2018), the researchers propose Stylometric Multitask Finetuning, a finetuning method that involves the transformer-based model to finetune to both classification and language modeling at the same time, allowing it to adapt to the stylometry of the data while learning to classify.

An augmentation to the classification head is done such that it could output two sets of logits, which is then interpreted as softmax probabilities for the next-word prediction task (language modeling) and softmax probabilities for the label classification.

Two losses from this "multitasking head" are then combined to be optimized together using a standard optimizer like Adam (Kingma & Ba, 2014).

Chapter 4

Methodology

This chapter outlines the general methodology of the thesis. It shows how each model was built and initialized, as well as the preprocessing decisions made with regards to the data and corpora used in each experiment.

4.1 The Fake News Dataset

A dataset was constructed composed of 3,206 news articles, each labeled *real* or *fake*, with a perfect 50/50 split between 1,603 real and fake articles, respectively. *Fake* articles were sourced from online sites that were tagged as *fake news sites* by the non-profit independent media fact-checking organization Verafiles[1] and the National Union of Journalists in the Philippines (NUJP)[2]. *Real* articles were sourced from mainstream news websites in the Philippines, including Pilipino Star Ngayon[3], Abante[4], and Bandera[5].

For preprocessing, only tokenization was performed on the dataset. The researchers employ a special type of tokenization called "Byte-Pair Encoding," (Sennrich et al., 2016) a form of fixed-vocabulary *subword tokenization* that considers *subword units* as the most primitive form of entity (i.e. a *token*) instead of canonical words (i.e. "I am walking today" *!* "I am walk ##ing to ##day"). Byte-Pair Encoding is useful due to a variety of reasons. First, it allows models to repre-

[1] vera les.org
[2] https://nujp.org/
[3] https://www.philstar.com/pilipino-star-ngayon
[4] https://www.abante.com.ph
[5] https://bandera.inquirer.net/balita

Split	Documents	Tokens	Unique Tokens	Num. of Lines
Training	120,975	39,267,089	279,153	1,403,147
Validation	25,919	8,356,898	164,159	304,006
Testing	25,921	8,333,288	175,999	298,974
OOV Tokens	28,469 (0.1020%)			

Table 4.1: Statistics for the WikiText-TL-39 Dataset.

sent out-of-vocabulary (OOV) words unlike standard tokenization, given that it treats every canonical character as a wordpiece by itself. Second, it helps language models in learning morphologically-rich languages as it now treats morphemes as primary enitites instead of canonical word tokens. For a more detailed exposition, the reader is directed to Cherry et al. (2018).

4.2 Wikitext-TL-39

To pretrain large language models for transfer learning, a large corpora is required. For this purpose, the researchers construct a large-scale preprocessed text corpora called WikiText-TL-39 (Cruz & Cheng, 2019). This dataset is used to pretrain the BERT models and the AWD-LSTM model used for later experiments. The corpus statistics for WikiText-TL-39 is shown on table 4.1.

4.2.1 Construction and Pre-processing

Since Tagalog Wikipedia does not have a list of verified "good" articles (Merity et al., 2016) and has far fewer content pages unlike its English counterpart (5,800,000 in English vs. 75,000), the proponents opted to instead scrape the content from all the listed pages in the Tagalog Wikipedia table of contents[6], narrowing down to just articles with titles that start with letters A-Z. Content was extracted using open-source Python packages Requests[7] and BeautifulSoup[8].

All characters were normalized into unicode and all HTML markup were unescaped. Normalization and tokenization were performed via the Moses Tokenizer (Koehn et al., 2007). The corpus was split into training, validation, and test sets with a ratio of 70%-15%-15%, respectively. When constructing the vocabulary, the researchers opted to not discard words that had a vocabulary count of less

[6]https://tl.wikipedia.org/wiki/Natatangi:Lahat_ng_mga_pahina
[7]https://pypi.org/project/requests/
[8]https://pypi.org/project/beautifulsoup4/

than 3, unlike in Chelba et al. (2013). This resulted in a vocabulary size of 279,153 tokens. All tokens in the test set unseen in the training set were replaced with special < unk > tokens.

4.2.2 Model-speci c Pre-processing

Pretraining with BERT requires a trained WordPiece vocabulary. The researchers opted to use the Byte-Pair Encoding (BPE) (Sennrich et al., 2016) model in Google's SentencePiece[9] library to train a custom vocabulary as Google did not release the original WordPiece code due to it having dependencies with their own internal libraries.

A xed vocabulary of 30,000 tokens for generating wordpieces were used in accordance with the speci cations of Google's own BERT models[10]

For use in ULMFiT, a light preprocessing scheme was followed that involves converting all words to lowercase, with a special < maj > token added in front of words that originally start with a capital letter. Likewise, all unknown words were changed to the < unk > token, and the vocabulary was limited to the top 30,000 words.

4.3 Siamese Network Training

A siamese neural network was trained as our baseline in order to compare transfer learning techniques to models designed for low-resource settings. For each twin, 300 dimensions were used for the embedding layer and a hidden size of 512 were used for all hidden state vectors.

To optimize the network, a regularized cross-entropy objective was used of the following form:

$$L(x_1; x_2) = y(x_1; x_2) \log p(x_1; x_2) + (1 \quad y(x_1; x_2)) \log(1 \quad p(x_1; x_2)) + \quad \text{jwj}^2 \tag{4.1}$$

where $y(x_1; x_2) = 1$ when x_1 and x_2 are from the same class and 0 otherwise. The Adam optimizer (Kingma & Ba, 2014) was utilized with an initial learning rate of 1e-4 to train the network for a maximum of 500 epochs.

[9] https://github.com/google/sentencepiece
[10] https://github.com/google-research/bert

4.4 Pretraining

4.4.1 BERT Pretraining

The researchers pretrain BERT Base models with 12 layers, 768 neurons per hidden layer, and 12 attention heads (a total of about 110M parameters) on the prepared WikiText-TL-39 (Cruz & Cheng, 2019) corpus and SentencePiece vocaularies using Google's provided pretraining scripts[11].

During pretraining, the proponents also experimented in pretraining settings involving smaller and larger vocabulary sizes, first using the full vocabulary size of the corpus (290,000 tokens), and then testing against a limited vocabulary size akin to the original BERT specifications (30,000 tokens).

For the masked language model pretraining objective, the proponents follow original specifications and use a 0.15 probability of a word being masked. The maximum number of masked language model predictions was also set to the original 20. All models use a maximum sequence length of 128 and a batch size of 256. The researchers use a learning rate of 1e-4 for all models.

All models are pretrained on Google Cloud Compute Engine using Google's Tensor Processing Units (TPU) version 3.8.

4.4.2 GPT-2 Pretraining

For GPT-2, the researchers pretrain a GPT-2 Transformer model on the prepared WikiText-TL-39 text corpora using language modeling as its sole pretraining task, according to the specifications of Radford et al. (2019).

The researchers use an embedding dimension of 410, a hidden dimension of 2100, and a maximum sequence length of 256. The proponents use 10 attention heads per multihead attention block, with 16 blocks composing the encoder of the transformer. A dropout on all linear layers was used to a probability of 0.1. All parameters were initialized to a standard deviation of 0.02.

For training, the proponents use a learning rate of 2.5e-4 and a batch size of 32, much smaller than BERT considering the large size of the model. The model was trained for 200 epochs with 1,000 steps of learning rate warmup using the Adam optimizer. The model was pretrained for 78 hours on a machine with one

[11] https://github.com/google-research/bert

NVIDIA Tesla V100 GPU.

4.4.3 AWD-LSTM Pretraining

For ULMFiT, the proponents train an AWD-LSTM language model using the prepared WikiText-TL-39 corpus. A 3-layer model was trained that uses an embedding size of 400 and a hidden size of 1150. The proponents set the dropout values for the embedding, the RNN input, the hidden-to-hidden transition, and the RNN output to (0.1, 0.3, 0.3, 0.4) respectively. A weight dropout of 0.5 was also used on the LSTM's recurrent weight matrices.

The model was trained for 30 epochs with a learning rate of 1e-3, a batch size of 128, and a weight decay of 0.1. The Adam optimizer was used slanted triangular learning rate schedules were utilized to anneal the learning rate during training. The model on a machine with one NVIDIA Tesla V100 GPU.

4.5 Finetuning

4.5.1 Standard Classi er Finetuning

The proponents netune the pretrained models to the target fake news classi cation task by preserving the pretrained weights and adding an appended classi cation layer or head

For BERT, the proponents append a classi cation head composed of a single linear layer followed by a softmax transformation to the transformer model. The BERT-Base model was then netuned on the fake news classi cation task for 3 epochs, using a batch size of 32 and a learning rate of 2e-5.

For GPT-2, a classi cation head is rst created, comprised of a layer normalization transform, followed by a linear layer, then a softmax transform. The pretrained GPT-2 transformer is then netuned for 3 epochs, using a batch size of 32 and a learning rate of 3e-5.

For ULMFiT, the proponents perform language model netuning on the fake news dataset (appending no extra classi cation heads yet) for a total of 10 epochs using a learning rate of 1e-2, a batch size of 80, and weight decay of 0.3. For the nal ULMFiT netuning stage, the proponents append a compound classi cation

head (linear! batch normalization! ReLU! linear! batch normalization! softmax).

The proponents then netune for 5 epochs, gradually unfreezing layers from the last to the rst until all layers are unfrozen on the fourth epoch. A learning rate of 1e-2 was used, and Adam's and parameters were set to 0.8 and 0.7, respectively.

4.5.2 Multitask Finetuning

To show the e cacy of Stylometric Multitask Finetuning, BERT and GPT-2 were both augmented to use this netuning setup with their classi cation heads. The proponents netune both models to the target task for 3 epochs, using a batch size of 32, and a learning rate of 3e-5. For optimization, the Adam optimizer was used with warmup steps of 10% the number of steps comprising 3 epochs.

Chapter 5

Results and Discussion

This chapter outlines the results for all experiments done over the course of the thesis.

5.1 Pretraining Results

For BERT pretraining, the proponents were able to train eight models, varying across vocabulary size, casing, and pretraining steps. The best uncased model (reaching a final loss of 0.0935) was trained for 500K steps with 5K steps of finetuning on the smaller 30K SentencePiece vocabulary, for a total of 33 hours. The best cased model (reaching a final loss of 0.0642), on the other hand, needed 1M pretraining steps with 10K warmup steps on the same 30K SentencePiece vocabulary, for a total of 66 hours. The proponents surmise that this is due to the model needing more steps to learn and get accustomed to casing.

The full results of BERT pretraining can be found on Table 5.1.

For ULMFiT, our AWD-LSTM language model reached a final validation loss of 4.4857 (which equals to 1.5009 perplexity). The model finished training for 30 epochs after around 11 hours.

For GPT-2, the GPT-2 Transformer model reached a final validation loss of 1.9820 (which equals around 7.2572 perplexity). The model finished training for 1000000 steps after around 78 hours.

Steps / Warmup	Casing	Vocab Size	Loss	MLM Acc	NSP Acc
500K / 5K	Cased	290K	0.3198	0.9158	0.9950
500K / 5K	Cased	30K	0.1046	0.9865	1.0000
500K / 5K	Uncased	290K	0.3396	0.9176	0.9986
500K / 5K	Uncased	30K	0.0935	0.9862	1.0000
1M / 10K	Cased	290K	0.1607	0.9563	0.9988
1M / 10K	Cased	30K	0.0642	0.9971	1.0000
1M / 10K	Uncased	290K	0.0716	0.9965	1.0000
1M / 10K	Uncased	30K	0.2600	0.9426	1.0000

Table 5.1: BERT Pretraining Results. MLM Acc refers to Masked Language Modeling objective accuracy. NSP Acc refers to Next Sentence Prediction objective accuracy. Figures inbold pertain to the best performing cased and uncased models.

5.2 Classification Results

The baseline siamese recurrent network achieved an accuracy of 77.42% on the test set of the fake news classification task.

The transfer learning methods gave comparable scores. BERT finetuned to a final 87.47% accuracy, a 10.05% improvement over the siamese network's performance. GPT-2 finetuned to a final accuracy of 90.99%, a 13.57% improvement from the baseline performance. ULMFiT finetuning gave a final accuracy of 91.59%, an improvement of 14.17% over the baseline Siamese Network.

It could be seen that transfer learning techniques outperformed the siamese network baseline, which the proponents hypothesize is due to the intact pretrained knowledge in the language models used to finetune the classifiers. The pretraining step aided the model in forming relationships between text, and thus, performed better at stylometric based tasks with little finetuning.

The model results are all summarized in table 5.2.

5.3 Multitask Finetuning Results

For Stylometric Multitask Finetuning, the proponents first outline empirical results that motivate the use of such a technique, then present results that show that transfer learning techniques can achieve better performance using the new finetuning method.

Model	Val. Acc.	Loss	Val. Loss	Pre. Time	Fine. Time
Siamese Net	77.42%	0.5601	0.5329	N/A	4m per epoch
BERT	87.47%	0.4655	0.4419	66 hours	2m per epoch
GPT-2	90.99%	0.2172	0.1826	78 hours	4m per epoch
ULMFiT	91.59%	0.3750	0.1972	11 hours	2m per epoch

Table 5.2: Final model results. Pretraining time refers to the number of hours the model took to nish the pretraining objective (masked-language modeling and next-sentence prediction for BERT, and language modeling for GPT-2 and ULMFiT (AWD-LSTM), respectively. Finetuning time refers to minutes per epoch. BERT and GPT-2 were netuned for 3 epochs, while ULMFiT was netuned for 5.

Model	Val. Accuracy
With LM Finetuning	91.59%
Without LM Finetuning	78.11%

Table 5.3: ULMFiT results with and without language model netuning. Removing the language model netuning step shows a signi cant drop in performance, giving evidence to the hypothesis that such a step improves the model by adapting to its stylometry.

5.3.1 Language Model Finetuning Signi cance

One of the most surprising results is that BERT and GPT-2 performed worse than ULMFiT in the fake news classi cation task despite being deeper models capable of more complex relationships between data.

The researchers hypothesize that ULMFiT achieved better accuracy because of its additional language model netuning step. The proponennts provide evidence for this assumption with an additional experiment that shows a decrease in performance when the language model netuning step is removed, droppping ULMFiT's accuracy to 78.11%, making it only perform marginally better than the baseline model. Results for this experiment are outlined in Table 5.3

In this netuning stage, the model is said to \adapt to the idiosyncracies of the task it is solving" (Howard & Ruder, 2018). Given that the transfer techniques rely on linguistic cues and features to make accurate predictions, having the model adapt to the stylometry or \writing style" of an article will therefore improve performance.

Model	Accuracy	Loss	Val. Loss
ULMFiT	91.59	0.3750	0.1972
BERT	91.20%	0.3115	0.3023
GPT-2	96.28%	0.2609	0.2197

Table 5.4: ULMFiT compared to transfer learning techniques augmented with stylometric multitask netuning. Including a language modeling netuning task to the transformer-based transfer learning techniques improved their performance, with GPT-2 outperforming ULMFiT by 4.69%. \Val. Accuracy" in this table refers to validation accuracy at test time.

5.3.2 Multitask-based Finetuning

The proponents used the Stylometric Multitask Finetuning technique over the standard netuning steps for BERT and GPT-2, motivated by the advantage that language model netuning provides to ULMFiT, and found that it greatly improves the performance of our models.

BERT achieved a nal accuracy of 91.20%, now marginally comparable to ULMFiT's full performance. GPT-2, on the other hand, netuned to a nal accuracy of 96.28%, a full 4.69% improvement over the performance of ULMFiT. This provides evidence towards the hypothesis that a language model netuning step will allow transformer-based transfer learning techniques to perform better, given their inherent advantage in modeling complexity over more shallow models such as the AWD-LSTM used by ULMFiT. Results for this experiment are outlined in table 5.4.

5.4 Transfer Learning Tasks Compared

Without the multitask-based netuning, ULMFIT o ers the advantage that it adapts to the stylometry of the target task. This allows it to perform incrementally better than BERT, despite it having a more shallow base language model (an AWD-LSTM with 3 layers, compared to a BERT-Base with 12 layers). It could also be seen that GPT-2 performs worse, albeit incrementally, to ULMFiT. The proponents hypothesize that GPT-2, despite lacking a language modeling netuning step, performed marginally close to ULMFiT and incrementally better then BERT due to a few architectural reasons, likely due to the depth and modeling exibility that GPT-2 has.

Compared to BERT, GPT-2 attends to 10 positions over 256 tokens at once, compared to BERT's 128 positions, which allows it to form relationships over long-term dependencies easier. This capability is important, especially in articles, where references to certain entities may have distances of paragraphs apart. Moreover, GPT-2 has 16 layers (or blocks) with 2100 hidden units each, as compared to BERT having 12 blocks with 768 units. This gives GPT-2 more exibility in terms of modeling the language it is being trained on. This poses GPT-2 at an advantage to BERT, which also makes it a good contender to ULMFiT.

With the multitask-based netuning step however, both models perform significantly better. BERT now outperforms ULMFiT marginally, and GPT-2 outperforms it with a good margin. Leveraging the ability to adapt to the stylometry of the target task gave the two transformer-based models better performance.

ULMFiT, however, is not without its advantages. On a computational level, ULMFiT is at an advantage with the fact that it requires less computational resources to pretrain.

BERT pretraining is expensive and requires more compute resources, with the authors recommending at least one Cloud TPU to pretrain a base model.[1] GPT-2 pretraining is likewise expensive, albeit requiring less memory bandwidth for data as it only uses one pretraining task, which makes GPUs available for use with it. However, it would take a GPU far more time to nish pretraining a GPT-2 compared to a TPU.

Pretraining an AWD-LSTM language model for ULMFiT, on the other hand, can be easily done in one relatively-modern GPU. It also requires much less time to pretrain (ours nishing in 11 hours) compared to a BERT model (ours nishing in 66 hours) and a GPT-2 model (ours nishing in 78 hours). In most low-resource, low-compute cases, ULMFiT is more advisable given its more accessible requirements.

5.5 Pretraining Tasks

All the transfer learning techniques were pretrained with a language modeling-based task. An AWD-LSTM and a GPT-2 transformer are basic language models trained to predict the next word. BERT, on the other hand, was pretrained using \masked language modeling" and next-sentence prediction. While language modeling has been empirically proven as a good pretraining task, we surmise that

[1] https://github.com/google-research/bert#pre-training-tips-and-caveats

other pretraining tasks could replace or support it.

Since automatic fake news detection uses stylometric information (writing style, language cues), we predict that the task could bene t from pretraining objectives that also learn stylometric information such as authorship attribution.

5.6 Generalizability Across Domains

When testing on three di erent types of articles (Political News, Opinion, Show-business/Gossip), the proponents nd that writing style is a prominent indicator for fake articles, supporting previous ndings regarding writing style in fake news detection (Potthast et al., 2017).

In general, news articles (political news included) tagged as real have a more \neutral stance" of reporting. Articles that are \persuasive" were more likely to be tagged as fake. This is a trend also followed by insinuative articles. In terms of showbusiness/gossip articles, articles reporting on celebrities were generally tagged as real news, following the trends with news in general. Gossip articles, especially \blind item" (gossip where the name of the subject is hidden) articles were always tagged as fake.

Here are a few examples of classi cations done on articles taken outside of the training and testing splits of the dataset that show the behavior of the classi er.

Pinagkakalaat na naman ng Rappler ang tungkol sa diumanoy Chinese trolls ni SAP Bong Go. Gusto nyo ba malaman ang katotohanan sa mga paratang na ito? Alamin dito sa Mindavote Live "Made in China Troll" episode. Classi er Label: Fake Gold label: Fake

Figure 5.1: Example of a fake article from a known fake news source. The article is easily identi able as fake given its straightforward, insinuative writing style, which is not present in mainstream real news articles.

Inaresto ng anti-scalawag and intelligence units ng Philippine National Police (PNP) ang isang anti-drug operative ng Pasay City Police Station ngayong araw, matapos na isangkot sa extortion at kidnapping.

Dinampot si Police Corporal Anwar Nasser, nakatalaga Station Drug Enforcement Team (SDET) ng Pasay police, ng mga tauhan ng PNP-Counter Intelligence Task Force (CITF) at Intelligence Group (IG).

Samantala, tatlo sa kanyang mga kasabwat ang nakatakas. Kinilala ang mga ito na sina Police Lieutenant Ronaldo Frades, hepe ng Pasay SDEU; Patrolman Anthony Fernandez; at Sergeant Rigor Octaviano, na pawang miyembro ng Pasay SDET.

Nag-ugat ang operasyon ng iulat ng isang alyas Joan sa PNP-CITF na si P/Cpl. Nasser ay nanghingi ng pera sa kanya kapalit ng paglaya ng kanyang live-in partner na si alyas George.

Sa report mula kay Police Colonel Romeo Caramat, head ng PNP-CITF, inaresto si George ng Pasay SDET sa Bautista Street, Makati City sa ganap na 9:00 ng umaga nitong Martes. Ayon sa mga pulis, nakuhanan ng droga si George sa operasyon.

Sinabi ni Joan na tinawagan siya ni P/Cpl. Nasser at hiningan ng P100,000 kapalit ng paglaya ni George.

Gayunman, sa halip na sumunod kay P/Cpl. Nasser, nagpasaklolo si Joan sa CITF upang masagip ang kanyang live-in partner.

Nagsagawa ang PNP-CITF at IG ng entrapment operation laban kay P/Cpl. Nasser at mga kasabwat nito sa loob ng Pasay SDET o ce nitong Martes, dakong 4:15 ng madaling araw.

Ayon sa CITF, nang tanggapin ng mga suspek ang P100,000 boodle money, naglabasan ang mga awtoridad.

Gayunman, si P/Cpl. Nasser lamang ang naaresto habang ang kanyang mga kasabwat ay nakatakas tangay ang boodle money.

Classi er Label: Real
Gold label: Real

Figure 5.2: Example of a news article from a mainstream news source. Identi able as such by its unbiased writing style.

Puro mapapaklang komento ang ipinakakain ngayon sa isang pamosong female personality ng mismong mga tagahanga niya. Pinasukan ng kawalan ng utang na loob at pangtatraydor pa nga ang kanilang emosyon.

Sila raw ang dahilan kung bakit natupad ang pangarap ng babaeng personalidad na sumikat, mas nadagdagan daw ang naiipon niyang pera dahil sa kanyang mga tagasubaybay, pero ano ang ginawa sa kanila?

Kuwento ng aming source, Iyakan sila nang iyakan! Sana raw, e, pinatay na lang sila ng idolo nila para isahan na lang ang sakit na naramdaman nila! Tinatawag siyang walang utang na loob, nagkakandagutom daw sila para sa pag-suporta sa idol nila, wala naman silang hinihinging kapalit, pero ang napala nila?

Hindi makakain ng kahit asong gutom ang mga salitang ibinabato sa kanya ngayon. Kung nakamamatay lang ang mga salitang nanunugat, e, baka pinag-buburulan na ngayon ang girl, simulang kuwento ng aming impormante. At kung galit sila sa babae ay triple nun ang galit nila sa aktor na piniling makare-lasyon ng kanilang idolo. Ipinaglihi raw sa manggang hinog ang male personality.

Kuwento uli ng aming source, Kanila raw ang huling halakhak kapag hindi nagtagal ang relasyon nila nung male personality. Halatang-halata raw naman kasing gagamitin lang niya ang girl.

Trophy girlfriend lang kuno niya ang babaeng karelasyon niya ngayon. Hindi raw naman kasi pambida ang lalaki, hanggang pagiging kontrabida na lang, kaya umaasa raw ang guy na baka maging bida siya kapag gumawa na sila ng pelikula ng female personality.

At ang nakakaloka pa, e, idinadamay nila sa issue pati ang mommy ng guy! Tuwang-tuwa raw ang nanay niya dahil ang girlfriend niya, e, sikat! Hindi raw magkandaugaga yung nanay ng male personality sa pag-aasikaso sa girl kapag nagpupunta sa bahay nila!

Classi er Label: Fake
Gold label: {

Figure 5.3: Example of a gossip (blind item) article from a mainstream tabloid. Notice that the classi er labels it as fake due to its writing style, despite not being inherently fake news.

NANININDIGAN ang Malacaang sa desisyon na payagan ang Department of Interior and Local Government (DILG) na ilabas ang pangalan ng mga kandidato na sangkot sa illegal drugs. Para sa Malacaang, makatutulong sa mga botante ang paglalabas ng narcolist para makapamili ng iluluklok sa puwesto. Handa naman daw ang Malacaang sa mga magsasampa ng kaso kaugnay sa paglalabas ng narcolist. Karapatan daw ito ng sinuman. Mabuti nga iyon para malinis ang pangalan.

Maraming bumabatikos sa idea ng DILG at Philippine Drug Enforcement Agency (PDEA) na ilabas ang mga pangalan ng kandidatong sangkot sa illegal drugs. Labag umano ito sa Konstitusyon. Ang nararapat daw gawin ng DILG o ng Malacaang ay sampahan ng kaso ang mga kandidatong ayon sa kanila ay sangkot sa illegal drugs. Maaaring magamit daw ito ng mga kalaban sa pulitika at paano kung hindi naman totoo ang paratang. Nasira na raw ang reputasyon ng kandidato at maaaring hindi na ito manalo. Sino pa ang boboto sa taong dinurog na ang pagkatao. Dapat daw ay mag-isip muna nang malalim bago ihayag ang mga pangalan ng kandidatong sangkot sa droga.

Pabor naman ang Philippine National Police (PNP) na ilantad na ang narcolist. Kung ano raw ang sabihin ni President Duterte ukol dito, susunod sila. Nasa tamang panahon naman ang paglalabas ng mga pangalan na makatutulong sa botante para makapamili.

Sa Marso 29 pa ang simula ng kampanya para sa local o cials at bago raw sumapit ang petsang ito ay ilalabas na ang narcolist. Marami umanong kandidato sa local posts ang sangkot sa droga kaya dapat nang ilabas ang listahan.

Siguruhin lamang ng DILG at PDEA na tama ang kanilang ilalabas na pangalan para maiwasan ang kaguluhan. Kumalap ng mga patunay na ang kandidatong ilalantad ang pangalan ay nakikinabang sa illegal na droga. Mahirap kapag inosente ang nailista.

Classi er Label: Real
Gold label: {

Figure 5.4: Example of an opinion article from a mainstream news organization. Notice that the classi er tags it as real due to its writing style despite being opinionated.

Chapter 6

Conclusion

In this thesis, the proponents have successfully performed the localization of fake news detection into the low-resourced Filipino language via multitask netuning.

They have shown that transfer learning techniques perform better than baseline few-shot techniques despite the latter being designed for low resources in mind. Moreover, they propose a new technique for netuning called stylometric multitask transfer that improves the performance of transformer-based transfer learning methods by a signi cant margin. They have also shown that the technique generalizes well and behaves accordingly to di erent types of articles, including straight news, celebrity news, gossip, and opinion articles.

For future work, the researchers propose the search for other stylometry-inherent pretraining tasks for use in settings where the downstream transfered task is stylometry-based, like fake news detection.

In addition, further re nements in terms of fake news detection could be added to tackle the task of fake newsveri cation , where a claim can be traced from its sources and then veri ed as fact or ction. This task is unsolvable using the techniques that are presented in this thesis, and as such, further study is needed to augment the work already done.

References

(n.d.-a).

(n.d.-b).

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. , abs/1409.0575. Retrieved from https://arxiv.org/abs/1607.06450

Bailey, K., & Chopra, S. (2018). Few-shot text classification with pre-trained word embeddings and a human in the loop. CoRR, abs/1804.02063.

Bourgonje, P., Moreno Schneider, J., & Rehm, G. (2017, September). From clickbait to fake news detection: An approach based on detecting the stance of headlines to articles. In Proceedings of the 2017 EMNLP workshop: Natural language processing meets journalism (pp. 84–89). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W17-4215 doi: 10.18653/v1/W17-4215

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a "siamese" time delay neural network. In Proceedings of the 6th international conference on neural information processing systems (pp. 737–744). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., & Koehn, P. (2013). One billion word benchmark for measuring progress in statistical language modeling. CoRR, abs/1312.3005. Retrieved from http://arxiv.org/abs/1312.3005

Cheng, C. (2018). A survey on data availability on veracles.

Cherry, C., Foster, G., Bapna, A., Firat, O., & Macherey, W. (2018, October-November). Revisiting character-based neural machine translation with capacity and compression. In Proceedings of the 2018 conference on empirical methods in natural language processing (pp. 4295–4305). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D18-1461

Cieri, C., Maxwell, M., Strassel, S., & Tracey, J. (2016, may). Selection criteria for low resource language programs. In N. C. C. Chair) et al. (Eds.), *Proceedings of the tenth international conference on language resources and evaluation (lrec 2016)*. Paris, France: European Language Resources Association (ELRA).

Conforti, C., Pilehvar, M. T., & Collier, N. (2018, November). Towards automatic fake news detection: Cross-level stance detection in news articles. In *Proceedings of the first workshop on fact extraction and VERification (FEVER)* (pp. 40–49). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W18-5507

Cruz, J. C. B., & Cheng, C. (2019). Evaluating language model finetuning techniques for low-resource languages.

Dekeersmaecker, J., & Roets, A. (2017). Fake news: Incorrect, but hard to correct. the role of cognitive ability on the impact of false information on social impressions. *Intelligence*, *65*, 107 - 110.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*. Retrieved from http://arxiv.org/abs/1810.04805

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*(Jul), 2121–2159.

Fakeblok.com. (2017). *Fakeblok: A journalist-moderated plug-in that flags fake news*. https://fakeblok.com/ .

Fe-Fei, L., et al. (2003). A bayesian approach to unsupervised one-shot learning of object categories. In *Computer vision, 2003. proceedings. ninth ieee international conference on* (pp. 1134–1141).

Feng, S., Banerjee, R., & Choi, Y. (2012). Syntactic stylometry for deception detection. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers - volume 2* (pp. 171–175). Stroudsburg, PA, USA: Association for Computational Linguistics.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, *abs/1703.03400*. Retrieved from http://arxiv.org/abs/1703.03400

Fiskkit.com. (2018). *Fiskkit: Discuss news that matters and find out what's true*. https://www.fiskkit.com/ .

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. CoRR, abs/1512.03385

Howard, J., & Ruder, S. (2018). Fine-tuned language models for text classi cation. CoRR, abs/1801.06146 Retrieved from http://arxiv.org/abs/1801.06146

Jin, Z., Cao, J., Guo, H., Zhang, Y., Wang, Y., & Luo, J. (2017). Rumor detection on twitter pertaining to the 2016 U.S. presidential election. CoRR, abs/1701.06250

Jin, Z., Cao, J., Zhang, Y., & Luo, J. (2016). News veri cation by exploiting con icting social viewpoints in microblogs. In Proceedings of the thirtieth aaai conference on arti cial intelligence (pp. 2972{2978). AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=3016100.3016318

Karimi, H., Roy, P., Saba-Sadiya, S., & Tang, J. (2018, August). Multi-source multi-class fake news detection. In Proceedings of the 27th international conference on computational linguistics (pp. 1546{1557). Santa Fe, New Mexico, USA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/C18-1131

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. CoRR, abs/1412.6980

Koch, G. (2015). Siamese neural networks for one-shot image recognition..

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., ... Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the acl on interactive poster and demonstration sessions (pp. 177{180). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1557769.1557821

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., J. Bethard, S., & McClosky, D. (2014, 01). The stanford corenlp natural language processing toolkit.. doi: 10.3115/v1/P14-5010

McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. CoRR, abs/1806.08730 Retrieved from http://arxiv.org/abs/1806.08730

Merity, S., Keskar, N. S., & Socher, R. (2017). Regularizing and optimizing LSTM language models. CoRR, abs/1708.02182 Retrieved from http://arxiv.org/abs/1708.02182

Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2016). Pointer sentinel mixture models. CoRR, abs/1609.07843. Retrieved from http://arxiv.org/abs/1609.07843

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Pennington, J., Socher, R., & Manning, C. D. (2015). Glove: Global vectors for word representation, 15321543..

Perez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017). Automatic detection of fake news. CoRR, abs/1708.07104. Retrieved from http://arxiv.org/abs/1708.07104

Politifact.com. (2018). The principles of the truth-o-meter: Politifacts methodology for independent fact-checking. http://www.politifact.com/truth-o-meter/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/#Truth-O-Meter\%20ratings.

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2017). A stylometric inquiry into hyperpartisan and fake news. CoRR, abs/1702.05638.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. Retrieved from https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners.

Ruchansky, N., Seo, S., & Liu, Y. (2017). Csi: A hybrid deep model for fake news detection. In Proceedings of the 2017 acm on conference on information and knowledge management (pp. 797–806). New York, NY, USA: ACM. doi: 10.1145/3132847.3132877

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Li, F. (2014). Imagenet large scale visual recognition challenge. CoRR, abs/1409.0575. Retrieved from http://arxiv.org/abs/1409.0575

Sennrich, R., Haddow, B., & Birch, A. (2016, August). Neural machine translation of rare words with subword units. In Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers) (pp. 1715–1725). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P16-1162 doi: 10.18653/v1/P16-1162

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017, September). Fake news detection on social media: A data mining perspective. SIGKDD Explor. Newsl., 19(1), 22–36. doi: 10.1145/3137597.3137600

Smith, L. N. (2015). No more pesky learning rate guessing games. CoRR, abs/1506.01186. Retrieved from http://arxiv.org/abs/1506.01186

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In Advances in neural information processing systems (pp. 4077–4087).

Triantafillou, E., Zemel, R., & Urtasun, R. (2017). Few-shot learning through an information retrieval lens. In Advances in neural information processing systems (pp. 2255–2265).

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. CoRR, abs/1609.03499

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need.

Vinyals, O., Blundell, C., Lillicrap, T. P., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. CoRR, abs/1606.04080.

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. Science, 359(6380), 1146–1151. doi: 10.1126/science.aap9559

Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., & Fergus, R. (2013, 17–19 Jun). Regularization of neural networks using dropconnect. In S. Dasgupta & D. McAllester (Eds.), Proceedings of the 30th international conference on machine learning (Vol. 28, pp. 1058–1066). Atlanta, Georgia, USA: PMLR. Retrieved from http://proceedings.mlr.press/v28/wan13.html

Wang, W. Y. (2017, July). "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) (pp. 422–426). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P17-2067 doi: 10.18653/v1/P17-2067

Wu, Y., Agarwal, P. K., Li, C., Yang, J., & Yu, C. (2014, March). Toward computational fact-checking. Proc. VLDB Endow., 7(7), 589–600. doi: 10.14778/2732286.2732295

Yu, M., Guo, X., Yi, J., Chang, S., Potdar, S., Cheng, Y., ... Zhou, B. (2018). Diverse few-shot text classification with multiple metrics. arXiv preprint arXiv:1805.07513.

Chapter 7

Research Ethics Documents

l </<Ÿ 1[Ql ghQs	.lhl<gEP jPQEh.lpQq]ZZQjll .lhl<gEP jPQEh$NMEl•Ä l[gs/s/g• <Ÿ l </<Ÿ 1[Ql ghQs! <[Q ÃÁÁÂ0<Ñ pl[kl•! <[Q<ÂÁÁÅ•+PQQdQlh . $³ G•k•lG<•dP¥çÄÃ\ÆÃÁÇÂÅ¥E÷ÆÃÄ	/$+ "]•• Ã]gZ "]•• Ã¥ ¦ 6lghQ]["]•• Â 6lghQ][<jl• !<

FRQILGHQWLDOLW\ DQG SURWHFW W YHUVLRQV RI WKH FRQVHQW IRUPV I			
:LOO GDWD FROOHFWLRQ LQYRO ,I <(6« OHW WKH SDUWLFLSDQW VLJ WKDW DVVXUHV WKDW KLV KHU DFD E\ SDUWLFLSDWLRQ RU QRQ SDUWLF			
:LOO GDWD FROOHFWLRQ LQYROYI YXOQHUDEOH JURXS 3:'V PLQRUL HWF ,I <(6« HQVXUH WKDW WKH SDUWLFL VWXG\ DQG WKH GDWD JDWKHULQJ S VSHFLDO DVVLVWDQFH VKRXOG WKH FRUUHVSRQGLQJ GHWDLOHG UHVHDU			
	<HV	1R	1RW $SSOLF
:LOO GDWD EH FROOHFWHG XVLQJ KDUGZDUH VRIWZDUH RU PHWD V ,I <(6«SURYLGH GHWDLOV RQ WKH D FUDZOHU HWF			
:LOO WKHUH EH LQIRUPDWLRQ DEI GHOLEHUDWHO\ ZLWKKHOG IURP W ,I <(6« ZKDW LQIRUPDWLRQ ZLOO E EHQHILWV IRU GRLQJ VR DQG KRZ Z			
:LOO \RXU GDWD EH VWRUHG DIWH ,I <(6 SOHDVH LQGLFDWH LQ WKH U ZLOO EH VWRUHG DQG KRZ LW ZLO LQIRUPDWLRQ LQ WKH LQIRUPHG FR IRU GDWD FRQILGHQWLDOLW\			
:LOO WKH GDWD EH PDGH DYDLOD			

l </<Ÿ 1[Ql ghQs	.lhl<gEP jPQEh.lpQq]ZZQjll .lhl<gEP jPQEh$NQEl•Ä l[gs/s/g• <Ÿ l </<Ÿ 1[Ql ghQs! <[Q ÃÁÁÂ0<Ñ pl[kl•! <[Q<ÂÁÁÅ•+PQQdQlh . $³ G•k•lGk•dP¥ÇÄÄ¡ÆÃÝÅÇÂÂ¥E÷ÆÃÄ	/$+ "]•• Ã]gZ "]•• Ã¥ ¦ 6IghQ]["]•• Â 6IghQ][<jl• !<

,I <(6«LQGLFDWH LQ WKH LQIRUPHG UHVXOWV RI WKH VWXG\ ZLOO EH X\ WKH LGHQWLW\ RI WKH SDUWLFLSDQ VWXGLHV 3URYLGH SDUWLFLSDQWV UHVXOWV WR EH XVHG IRU IXUWKHU ZLWKGUDZ WKHLU UHVXOWV IURP IX			
:LOO \RX DQRQ\PL]H WKH SDUWLFI ,I <(6&URYLGH GHWDLOV RQ KRZ \RX FRQILGHQWLDOLW\ DQG DQRQ\PLW\			
:LOO WKH UHVXOWV RI WKLV VW ,I \HV GR \RX LQWHQG WR DSSO\ IR UHVHDUFK" 3OHDVH FKHFN BBBBB <HV BBBBB 1R			

. .]gZ "]•Ã¥¦ /0 "0." 0./ .

RESEARCH ETHICS CLEARANCE FORM For Thesis Proposals[1]	
Names of student researcher/s :	*Jan Christian Blaise Cruz* *Julianne Agatha Tan*
College:	College of Computer Studies
Department:	Software Technology Department
Research Title:	Localization of Fake News Detection via Multitask Transfer Learning
Course:	BS Computer Science with specialization in Software Technology
Expected duration of project:	from: T3 AY17-18 to: T3 AY18-19
Ethical considerations *Wikipedia articles and newspaper articles that were crawled do not contain identifiable information such as metadata.*	

To the best of our knowledge, the ethical issues listed above have been addressed in the research.

Charibeth Cheng

Name and signature of adviser/mentor

Date:

Name and signature of panelist

Date:

Name and signature of panelist

Date:

[1]The same form can be used for the reports of completed projects. The appropriate heading need only be used.