EXPLOITING MULTILINGUAL CONTEXTUAL EMBEDDINGS FOR SINHALA TEXT CLASSIFICATION

G.V. Dhananjaya

218039D

Master of Science (Major Component Research)

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

May 2022

EXPLOITING MULTILINGUAL CONTEXTUAL EMBEDDINGS FOR SINHALA TEXT CLASSIFICATION

G.V. Dhananjaya

218039D

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science (Major Component Research)

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

May 2022

DECLARATION

I declare that this is my own work and this Thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 2023/04/20

The supervisors should certify the Thesis with the following declaration.

The above candidate has carried out research for the Master of Science (Major Component Research) Thesis under our supervision. We confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr. Surangika Ranathunga

Signature of the Supervisor: Surangika Ranathunga Digitally signed by Surangika Ranathunga Date: 2023.05.17 00:19:18 +12'00' Date:

Name of Supervisor: Prof. Sanath Jayasena

Signature of the Supervisor: Date: 22/05/2023

DEDICATION

I dedicate this research work to all the individuals who supported me in academics, including my family, friends, and teachers.

ACKNOWLEDGEMENT

I owe gratitude to my supervisors, Dr. Surangika Ranathunga and Prof. Sanath Jayasena for the immense support, guidance and supervision they provided to complete this work. I would also like to thank Mr. Piyumal Demotte who overtook the task of pre-training SinBERT language models.

ABSTRACT

Language models that produce contextual representations (or embeddings) for text have been commonly used in Natural Language Processing (NLP) applications. Particularly, Transformer based, large pre-trained models are popular among NLP practitioners. Nevertheless, the existing research and the inclusion of low-resource languages (languages that primarily lack publicly available datasets and curated corpora) in these modern NLP paradigms are meager. Their performance for downstream NLP tasks lags compared to that of high-resource languages such as English. Training a monolingual Language model for a particular language is a straightforward approach in modern NLP but it is resource-consuming and could be unworkable for a low-resource language where even monolingual training data is insufficient. Multilingual models that can support an array of languages are an alternative to circumvent this issue. Yet, the representation of low-resource languages considerably lags in multilingual models as well.

In this work, our first aim is on evaluating the performance of existing Multilingual Language Models (MMLM) that support low-resource Sinhala and some available monolingual Sinhala models for an array of different text classification tasks. We also train our own monolingual model for Sinhala. From those experiments, we identify that the multilingual XLM-R model yields better results in many instances. Based on those results we propose a novel technique based on an explicit cross-lingual alignment of sentiment words using an augmentation method to improve the sentiment classification task. There, we improve the results of a multilingual XLM-R model for sentiment classification in Sinhala language. Along the way, we also test the aforementioned method on a few other Indic languages (Tamil, Bengali) to measure its robustness across languages.

Keywords: Multilingual language models, Multilingual embeddings, Text classification, Sentiment analysis, Low-resource languages, Sinhala language

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
AP	Auxiliary phrases
CL	Continual Learning
MLM	Masked Language Modeling
MMLM	Multilingual Language Models
NER	Named Entity Recognition
NLG	Natural Language Generation
NLI	Natural Language Inference
NLP	Natural Language Processing
NLU	Natural Language Understanding
NSP	Next Sentence Prediction
POS	Part-of-Speech tagging
RNN	Recurrent Neural Network
TLM	Translation Language Modelling

CHAPTER 1

INTRODUCTION

Natural Language Processing (NLP) has a plethora of applications involving different aspects of human languages. Text classification is one of such primary tasks in NLP, where text is classified at token (word), sentence, or document level and falls into the broad sub-category of Natural Language Understanding (NLU) . Modern NLP heavily relies on Transformer [1] based language models. This is mainly due to the reason that Transformer based language models have been successful in creating meaningful, contextual vector representations for natural language text. Typically, such models are pre-trained on a large dataset using an unsupervised objective such as Masked Language Modeling (MLM) . BERT [3], RoBERTa [4], and T5 [5] are three popular examples of Transformer based language models that support English. Typically, a corpus of substantial size is required to pre-train such a model. The most frequently followed approach to utilize such a model in a downstream task such as text classification is fine-tuning the whole model using a dataset suited for the target task in a supervised manner. This paradigm has been applied to many benchmark datasets and tasks such as GLUE, XNLI [6, 7]. Despite these models showing state-of-the-art results for such benchmarks and datasets, their utilization of languages has mostly been confined to an elite set of high-resource languages with English being at the top of the list. This is due to that high-resource languages have the availability of abundant corpora and datasets where low-resource languages do not have such. As a result, performance for low-resource languages such as Sinhala (categorized as belonging to the extremely low-resource language class in Joshi's [8] language categorization) are subpar compared to the results of high-resource languages, even in the very few instances they have been experimented with. Monolingual models have shown some promising results but pre-training monolingual language models for resource-poor languages is not always feasible and such attempts consume cumbersome amounts of computational resources as well [9].

As a potential solution, MLMs have been pre-trained on multilingual datasets. They also offer the ability of transfer-learning across languages, such that low-resource languages could benefit from the pre-trained knowledge of the high-resource languages included in the pre-training stage of the model[1]. This has proven beneficial, especially for incorporating unseen low-resource languages (languages that were not a part of the pre-training stage of the model) for downstream NLP tasks [10]. mBERT, XLM-R and

[1] As per the common terminology, cross-lingual models are typically referred to as models having latent spaces where similar entities have similar/closer vector representations. Although multilingual models share a common latent space for multiple languages, they are not expected to possess the above characteristic. From here onward, the term multilingual models will be used collectively for models supporting multiple languages.

mT5 [3, 11, 12] are such MMLMs supporting around 100 languages in each model. Yet, resource-rich languages dominate the multilingual models as well, in terms of the pre-trained corpora used for training the models as well as the performance (10, 11). Hence, improving the model performance for low-resource languages in multilingual models should be carried out. This could be workable considering the transfer-learning ability offered by those multilingual models. The difficulties of pre-training new monolingual models for a low-resource language further support this idea.

1.1 Research problem

Several techniques have been proposed to improve the performance of multilingual models for low-resource language tasks. These include methods such as data augmentation [13], tokenizer optimization [14], multi-task fine-tuning [15, 16] and an ensemble of models. Additionally, the integration of external knowledge has been a promising path to improve Transformer model performances more recently. There, integrating external factual and/or linguistic knowledge into the model is carried out in addition to the already learned knowledge during the pre-training phase by the model. Mostly, these methods have been tested infrequently on low-resource language datasets and the Sinhala language is yet to be experimented with. Hence, it is required to inspect whether the viability of such methods that have been tried for other languages are valid for improving results (i.e.- text classification) for the Sinhala language. Additionally, it would equally be important and interesting to devise novel methods which suit low-resource languages such as Sinhala.

1.2 Research objectives

The main objectives of this research are,

1. Identifying the best performing Transformer based language models for Sinhala text classification, among the existing monolingual and multilingual models that support the Sinhala language.

2. Pre-train a new monolingual language model for Sinhala, which we coin as “Sin-BERT".

3. Propose a feasible, novel method that combines external knowledge with pre-trained language models for improving downstream sentiment analysis (a text classification task) results for the Sinhala language.

1.3 Contributions

1. Performing an extensive empirical study that compares the performance between existing multilingual and monolingual models, for an array of text classification tasks in the Sinhala language (sentiment analysis, news source classification, writing style classification, news category classification) - this is the first empirical study for Sinhala text classification.

2. We publicly release new annotated datasets for Sinhala news source classification, writing style classification, and news category classification.

3. We propose a novel method to improve the results of sentiment classification in low-resource languages. The proposed method is based on a high-resource language lexicon (as an external knowledge base) and a cross-lingual alignment of sentiment words (without a separate objective function).

Publication(s):

1. Full paper[2] published at LREC (Language Resources and Evaluation Conference, 13th edition) - 2022 (Marseille, France). *published* ISBN: 979-10-95546-72-6, ISSN: 2522-2686

   Title - *BERTifying Sinhala - A Comprehensive Analysis of Pre-trained Language Models for Sinhala Text Classification, Vinura Dhananjaya, Piyumal Demotte, Surangika Ranathunga and Sanath Jayasena* [17]

2. Journal Paper (Pending publication)

   Title - *Lexicon-based Fine-tuning of Multilingual Language Models for Low-resource Language Sentiment Analysis, Vinura Dhananjaya, Surangika Ranathunga, Sanath Jayasena*

Datasets/Models:

- We have released two pre-trained Sinhala monolingual models and classification datasets on - https://huggingface.co/NLPC-UOM

1.3.1 Thesis organization

We organize this thesis in the following way. The next chapter contains a literature survey where we have reported findings in the contemporary literature. The two chapters that entail it report the details of our experiments. In the final chapter, we present our conclusion and possible future work.

[2] https://arxiv.org/abs/2208.07864

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Overview

Following the contemporary use of Deep Learning models in NLP, in this work, Transformer based language models were primarily used for Sinhala text classification . In the following sub-sections, we present theoretical concepts related to the main paradigms of this research.

2.2 Sinhala Language

Sinhala is predominately spoken by the Sinhalese populace (17 million) in Sri Lanka and belongs to the broad Indo-Aryan language family. Similar to many other languages in this language family, Sinhala is an inflectional (or fusional) language, meaning that the language uses morphemes to modify a word (verb, nouns, adjectives etc.) by adding them to the root word (inflecting) to derive a new meaning/properties to the particular word. (e.g.- conjugation of verbs in English). This is opposed to agglutinative languages, which simply combine morphemes to form different words.
Moreover, Sinhala language contains other rich linguistic features, such as *diglossia*[1] (having two distinct systems for verbal communication and written/official communication) and at discourse level Sinhala is considered as *pro(noun)-drop*[2] which refers that some words in a sentence can be omitted (dropped). In addition to being a morphologically rich language, having a complex script, regional dialects and a diverse polysemous vocabulary which is also inspired by foreign languages amount to making Sinhala language posses immanent challenges for computational understanding of the language.

2.3 Transformer-based pre-trained models

The prime intention behind pre-trained language models is to effectively learn text representations at a large scale. Transformer architecture was proposed by Vaswani et al. [1] as a replacement for earlier used Recurrent Neural Network (RNN) based language models such as LASER [18] and ELMo [19]. Text data being sequences, a specific mechanism is required to understand the relations between distinct parts within a sequence. The RNN-based language models had a computational constraint, which is that RNN (and later architectures such as LSTM [20] and GRU [21]) limits to se-

[1] https://www.britannica.com/topic/diglossia
[2] https://doi.org/10.1093/acrefore/9780199384655.013.610

quential operations (hinder parallel processing). These earlier architectures also do not scale well to longer input text sequences due to an effect called “vanishing gradient". Figure 2.1 shows a traditional RNN mechanism[3]. X_t denotes the parts of the input sequence (e.g.- tokens in a text) and Y_t denotes the corresponding outputs. Note that for calculating the outputs, RNN model uses previous inputs through the intermediate states A_t.

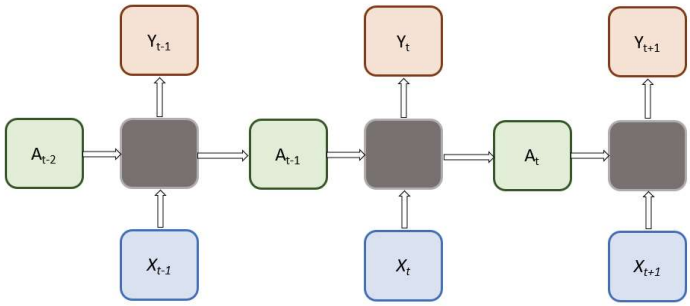

Fig. 2.1: Traditional RNN architecture

Nevertheless, the alternatives proposed to alleviate this issue bring about a limitation to learn dependencies between distant positions in a text sequence [22, 23]. The Transformer model utilizes *Self-attention* mechanism which takes account of different positions within a single sequence to calculate a vector representation for it. With this technique, Transformer architecture is able to handle longer text sequences well while handling the “vanishing gradient" issue better than RNNs and LSTM-based neural nets. Transformer also allows parallel processing on sequences making processing text more computationally efficient as well.

A Transformer-based language model typically consists of an encoder and a decoder module so does the Transformer architecture. However, Transformer based language models can utilize different choices of architectures for different downstream tasks. Encoder-decoder stack is used for Natural Language Generation (NLG) tasks such as machine translation and summarization, where the model should produce text sequences as the output. There are decoder-only models as well, such as GPT [24]. For NLU tasks such as text classification, encoder-only language models such as BERT are typically utilized. XLM, XLM-R, mBERT are such encoder-only models which are also multilingual. Hence, in our experiments, we specifically focus on encoder-only models.

[3]https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

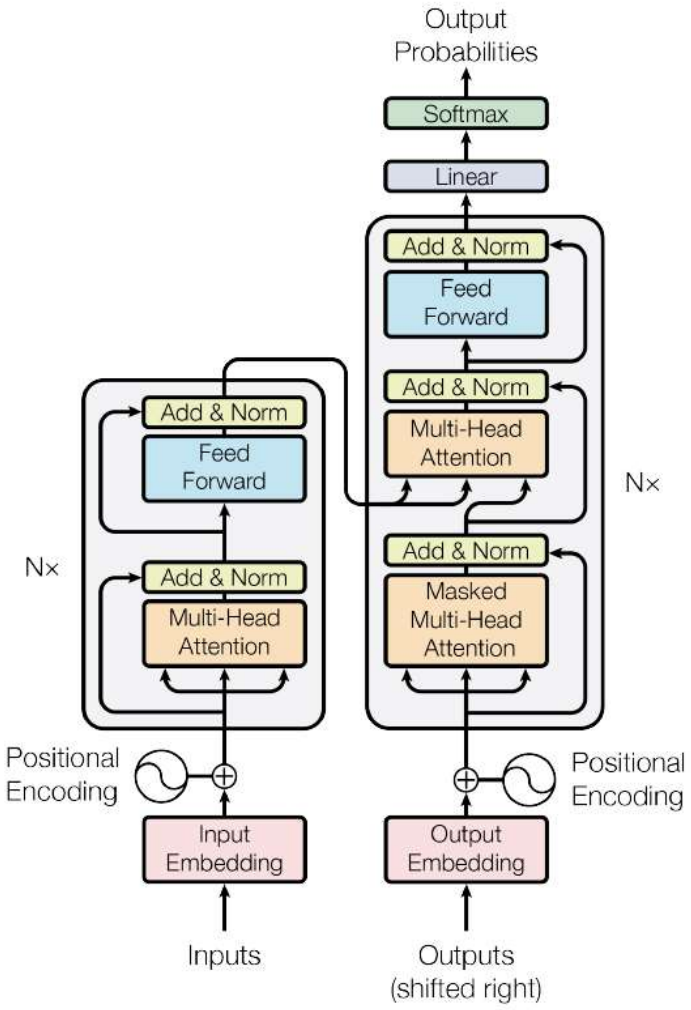

Fig. 2.2: Transformer model architecture [1]

2.3.1 Encoder of the Transformer

Usually, an encoder refers to a stack of encoder modules. Simply, the input text sequence is fed to the encoder which produces a set of hidden state vectors for them. These hidden representations are then fed into a decoder which predicts the probable tokens (e.g.- words) of the sequence based on the previously seen parts of the sequence.

$$X_{1:n} = \{x_1, x_2, ..., x_n\} \tag{2.1}$$

$$f_{\theta_{encoder}} : X_{1:n} \rightarrow \bar{X}_{1:n} \tag{2.2}$$

$$p_{\theta_{decoder}}(Y_{1:m}|\bar{X}_{1:n})) \tag{2.3}$$

$$p_{\theta_{encoder},\theta_{decoder}}(Y_{1:m}|X_{1:n})) \tag{2.4}$$

Equation 2.2 shows the mapping of an input sequence of tokens (e.g.- words) in the format shown in Eq. 2.1 to a sequence of hidden states. Then, as per Eq. 2.3, the decoder tries to model the conditional probability distribution of the output vector sequence based on the hidden states. As per Eq. 2.4, the congregation of encoder-decoder models the conditional probability distribution of a target text sequence, when an input text sequence is given.

Attention in a Deep-Learning model refers to a similar concept relevant to humans and other intelligent beings, which is about focusing on specific parts of the input data and emphasizing less on others. In NLP, attention is important to distinguish certain parts of a text sequence, which impact the overall meaning of the sentence. Self-attention - an improved form of attention, is a distinguishing part of the Transformer based language models when compared with RNN-based language models. Within the encoder blocks of the Transformer, it can help create contextual representations of input text sequences by forming relationships between every tokenized part of the text sequence and it is bi-directional as well; meaning the forming of relationships can be done along both directions of the text sequence. Self-attention is implemented in a way that contextual representations of text sequences can be formed by taking long-range dependencies into account while parallelizing the relevant mathematical operations.

$$X'_{1:n} = \{x'_1, x'_2, ..., x'_n\} \tag{2.5}$$

$$X''_{1:n} = V_{1:n}\{softmax(Q^T_{1:n}K_{1:n})\} + X'_{1:n} \tag{2.6}$$

Equation 2.5 depicts an input text sequence fed into a self-attention layer in the Transformer whereas Eq. 2.6 shows the operation of self-attention, where it creates a contextualized representation of the text sequence. There, ***K, Q, V*** represent *Key, Query* and *Value* matrices respectively, which are also learnable. The *softmax* operation creates self-attention weights used to calculate the output vectors of the particular self-attention layer. Self-attention layer is present in both encoder blocks and decoder blocks, with the difference that it is not bi-directional (only uni-directional; left-to-right) in decoder blocks.

Transformer models have shown the capability of efficiently learning contextual latent representations of text among them only through some unsupervised pre-training objectives. The most commonly used pre-training objective is MLM [3]. MLM masks several words in the unannotated pre-training corpora and imposes the model to predict the masked words (e.g.- BERT uses MLM by masking 15% of the tokens in the text sequence). Another one is Translation Language Modelling (TLM) where parallel source text and its translated text are fed into the model, by masking words in both source and target. Prominent Transformer based language models use one or a few

of such objectives. LaBSE [25], which is based on a dual-encoder architecture based on BERT, uses MLM and TLM as its pre-training objectives. XLM-R uses MLM as its only pre-training objective. Other different kinds of pre-training tasks have been utilized in Transformer models such as contrastive learning-based methods [26, 27], alignment-based methods [28] and cross-lingual MLM [29]. Qiu et al. [30] report an array of such pre-training objectives.

2.3.2 Transformer decoder module

As presented through Eq. 2.3, the decoder models a probability distribution that can be utilized to predict text sequences when an input sequence is given. Unlike the encoder, self-attention in the decoder is inhibited to be uni-directional. This is because the decoder is expected to predict parts of the output sequence based on previously seen and predicted parts of it. Equation 2.3 can be expanded using Bayes's theorem as a product of each single vector component of the expected target vector of the output text sequence as shown in Eq. 2.7. There, the 0^{th} target vector (Y_0) is derived from the representation of a "BOS" (Beginning Of Sentence, or [CLS] in some implementations) token added to the sequence, by the model.

$$p_{\theta_{decoder}}(Y_{1:m}|\bar{X}_{1:n})) = \prod_{i=1}^{m} p_{\theta_{decoder}}(y_i|Y_{0:i-1}, \bar{X}_{1:n})) \tag{2.7}$$

The predicted target sequence vectors ($Y_{1:m}$) then can be mapped with the model's vocabulary through softmax operations. There, each y_i gets a probability distribution from the softmax operation, which is depicted by the $p_{\theta_{decoder}}$ in Eq. 2.7.

2.3.3 Overall Transformer architecture

Figure 2.2 shows the original model architecture proposed in the revolutionary paper Vaswani et al. [1]. It comprises an encoder module and a decoder module which again includes 6 layers in each. In the encoder module, in each of the identical 6 layers (N), there is a multi-head attention sub-layer and a feed-forward neural network layer. Both these sub-layers are accompanied by a layer normalization and residual connection around them. In the decoder layer, the differences are there is a third sub-layer of multi-head attention and the self-attention mechanism is modified to access only the previous positions of a sequence. This is shown as *Masked multi-head attention* in the figure. For the multi-head attention, Transformer utilizes 8 parallel attention layers.
The feed-forward neural networks in the encoder or decoder layers in this Transformer architecture are *position-wise feed-forward networks* which are connected to every position of an input sequence identically. Although the paper mentioned a ReLU activa-

tion function is used in the feed-forward networks, some off-the-shelf implementations of Transformer based language models may utilize different activation functions as well (such as Huggingface's[4] models). Positional encodings preserve the information about the position of the tokens in the input sequence. Embeddings are responsible for vectorizing an input text sequence usually utilizing an embedding matrix. The embedding matrix maps tokens to pre-defined (learned) vectors. These vectors should be context-independent at first, the model then learns contextualized representations from the training data.

2.3.4 Tokenizers

Tokenizers are essential for every Transformer based NLP model as they act as the preliminary step to convert text from raw, human-understandable formats into a machine-readable mathematical format. Basically, it splits an input text into smaller parts (e.g.- sentence into words). These learned tokens are unified as the model's vocabulary. Based on how this splitting is executed, tokenizers can be categorized. Subword tokenizers are more popular than the other two types as they are more efficient. Rather than splitting at word level or character level, subword tokenizers tokenize text at a subword level. Moses[5] is a word-level tokenizer as well as a rule-based tokenizer. Dai et al. [31] also use a word-level tokenizer which results in a huge vocabulary.
Many Transformer based language models make use of versions of Subword tokenizers. Byte-Pair Encoding (BPE) [32] is a subword tokenizer which utilizes a word based or a rule-based tokenizer at a pre-tokenization step. Afterward, it learns to merge them and form a subword vocabulary. CONNEAU and Lample [33] and Liu et al. [4] use this tokenizer. Byte-Level BPE (bBPE) and Wordpiece are also based on BPE, whereas the Wordpiece tokenizer is used in the popular BERT model. Unigram [34] follows a contrasting approach when building its subword vocabulary compared to the BPE approach. It first constructs a larger vocabulary and learns subwords such that it reduces the size of the initial vocabulary. Sentencepiece tokenizers [35] are used in models such as Conneau et al. [11], which provides an effective solution to tokenize text of the languages that do not use spaces to separate words (e.g.- Chinese). Generally, it is used in conjunction with the Unigram tokenizers.

2.4 Transformer based deep-learning language models

In the subsequent subsections, we describe few of the Transformer based encoder models, which are also relevant to the work presented in this thesis.

[4]https://huggingface.co/
[5]http://www.statmt.org/moses

2.4.1 BERT

BERT [3] (Bidirectional Encoder Representations from Transformers) can be identified as the first Transformer based language representation model which is capable of building representations based on both directions (left-to-right, right-to-left) of a text sequence. This is fulfilled by the MLM pre-training objective. The other pre-training objective of BERT is Next Sentence Prediction (NSP) which checks whether two segments of text precede or succeed each other. BERT-base, the smaller variant of BERT (BERT-base) contains 12 Transformer layers (L), a hidden size (H) of 768, 12 self-attention heads (A), and 110 million model parameters. BERT-large contains L=24, H=1024, and A=16 with 340 million parameters. BERT uses BookCorpus as its pre-training dataset and it can be claimed that BERT laid foundations of many popular Transformer based language models introduced later.

2.4.2 mBERT

mBERT[6] [3], which stands for “multilingual BERT" followed the BERT model as its multilingual version. mBERT utilizes the same pre-training procedure in BERT and supports 104 languages. The original mBERT model architecture resembles the same architecture seen in the BERT-base model. In order to balance the pre-training multilingual dataset sizes, it uses oversampling and undersampling of corpora for low-resource and high-resource languages respectively.

2.4.3 RoBERTa

RoBERTa (Robustly Optimized BERT Approach) is an improvement of BERT that aims to enhance downstream task performance through some changes in the pre-training procedure introduced in BERT. The changes are mainly introduced as; 1. dynamic masking of MLM replacing the static masking pattern used in BERT, 2. removal of NSP loss and using full sentences to form NSP pre-training dataset, and 3. utilizing larger batch sizes with more data, and training the model for a longer period, which show improvements over BERT in downstream tasks. The model uses BookCorpus and Wikipedia data for pre-training the model. RoBERTa model architecture follows BERT-large model’s architecture.

2.4.4 XLM-R

XLM-R is a multilingual encoder-based model which has shown improved performance across several benchmarks surpassing mBERT and XLM [33] models. XLM-R has been pre-trained on 100 languages with the use of a new CommonCrawl dataset

[6]https://github.com/google-research/bert/blob/master/multilingual.md

(CC-100) in contrast to the Wikipedia data used by mBERT and XLM multilingual models. XLM-R also shows competitive performances compared to monolingual models such as RoBERTa in benchmarks such as GLUE and XNLI. It is a stark feature that XLM-R uses the MLM objective as its sole pre-training objective. Moreover, the authors of XLM-R bring about the argument that increasing the model capacity could assist the model to improve or maintain task performance while including more languages which XLM-R model fulfills. XLM-R Base model is parameterized as; L= 12, H = 768, A = 12, 270 million total parameters and XLM-R-large as L = 24, H = 1024, A = 16, 550 million total parameters.

2.4.5 LaBSE

LaBSE is a multilingual model based on BERT and has been devised specifically for downstream tasks such as bi-text mining and sentence similarity. LaBSE has a special dual-encoder architecture with encoders based on BERT (L=12, A=12, H=768) and it pre-trains this dual encoder model with an “additive margin softmax loss". LaBSE uses both monolingual data and bilingual translation pair datasets to pre-train the model. LaBSE is then tested on translation-related tasks such as Tatoeba[7] and also on zero-shot performance for the Tatoeba task for some low-resource languages. LaBSE also claims to have improved performance over mBERT on tasks like Tatoeba on low-resource languages. The reason for the improvement given as using a larger vocabulary, using Translation Language Modelling (TLM) in addition to MLM, and using the CommonCrawl dataset for pre-training.

2.4.6 LASER

LASER is also a multilingual model yet not based on Transformer architecture. (We describe the LASER model since we use it in our experiments to evaluate language models). It is based on Bi-LSTM (Bidirectional LSTM) which can gain information about a text sequence in both directions with the help of two LSTM networks. One of the main purposes of the LASER model has been intended to perform zero-shot Transfer Learning for a target language-task pair by only learning a header module on a related English dataset for a task such as classification. LASER model has been experimented with tasks such as bi-text mining and NLI [18]. Natural Language Inference (NLI) and cross-lingual document classification. For pre-processing the data, LASER relies on language-specific tools such as “Moses" unlike the later Transformer models which use language-agnostic tokenizers. However, the pre-trained Bi-LSTM encoder could be used for any target task without being fine-tuned (only a task specific header would need to be fine-tuned).

[7] https://tatoeba.org/

2.5 Utilizing multilingual models for target tasks

2.5.1 Vanilla fine-tuning

Transformer based neural models can be considered as task-agnostic language models which can be fine-tuned for a target task, usually with an annotated dataset smaller than the pre-training dataset. Generally, text classification refers to a supervised-learning task on an annotated dataset and predicting labels for data points unseen during the learning phase by the model. In modern NLP, text can be sized within a broad range, such as individual words, sentences, paragraphs and documents. Occasionally, the term document is used to identify paragraphs and much larger documents, interchangeably. Whenever sufficient annotated datasets are available for the target task, we can perform a standard fine-tuning (vanilla fine-tuning) of the model. For a classification task it would be training the model as a supervised learning task with a labelled dataset (e.g.- binary classification with binary cross entropy loss function).

2.5.2 Other fine-tuning methods

A frequently followed approach for text-classification of low-resource languages (or in instances where target task data is not adequate) is using zero-shot learning. There, the model is fine-tuned on the target task using a task-relevant high-resource language (such as English) dataset, and the model infers labels for the target low-resource language dataset without explicitly being fine-tuned for the target language [10, 36]. This approach would be restricted to multilingual models. There are several factors that affect the performance of a multilingual model: model capacity, amount of data used for pre-training, amount of fine-tuning data as well as tokenizer used in the model. These are further mentioned in Chapter 3. Moreover, Sun et al. [37] discuss methods to fine-tune a BERT model for target downstream classification tasks, which can be applied generally to other Transformer-based encoder models as well.
Apart from the vanilla fine-tuning method, there are several modified methods of fine-tuning. Typically, in standard fine-tuning, all the model parameters are updated through supervised learning, which is referred to as *global fine-tuning*. In *feature-based fine-tuning*, we can selectively update the parameters of the models. It is mentioned that the earlier method is more convenient and often successful than feature-based fine-tuning [30, 38]. There are other types of fine-tuning Transformer models that focus on using modified objective functions/loss functions. As an example, Wang et al. [39] and Li et al. [40] utilize “Instance weighting" for a cross-domain text classification task and a cross-lingual zero-shot classification task respectively. This is a method that weights under learned examples by calculating required weights based on the output of the model’s loss function and applying the weights when model parameters are

updated (i.e.- with mini-batch gradient descent function). Adding additional parameters to the model while keeping the model's original pre-trained parameters intact is a novel method of adapting a pre-trained model for a downstream task. This was proposed by Houlsby et al. [41].

There are persisting obstacles to fine-tuning of pre-trained models. *Catastrophic forgetting* is one such dilemma that simply refers to forgetting the already learned knowledge in a pre-trained model when that model is further fine-tuned. This has been frequently observed with paradigms such as *continual learning* [42], where a model tends to forget the knowledge from earlier stages or tasks as it continues to train on data with different distributions. A possible circumvention for this problem is feature-based fine-tuning, yet it has been observed as less effective than global fine-tuning (as mentioned above as well). Ermis et al. [43] suggest a new algorithm based on Adapters [2, 41] as a possible solution for *catastrophic forgetting* in a text classification task. There are several studies that propose solutions for catastrophic forgetting problems such as Ke et al. [44] and Ke et al. [45] which incorporate continual learning for sentiment analysis tasks. Discriminative fine-tuning is another approach of fine-tuning where varying learning rates are applied at each layer of the model. This idea was first introduced by [46] and they also introduce methods such as gradual unfreezing (the number of trainable model parameters is increased gradually) which are also intended as means to overcome catastrophic forgetting. However, they experiment with an LSTM-based model. It has also been observed that different layers of pre-trained language models capture different information from the training [37] thus making it challenging to find the optimum output layer for a given task.

2.5.3 Transfer Learning in Transformer models

Large Deep Learning models based on Transformers demand substantial datasets to yield satisfactory results which are hard to achieve especially with low-resource languages and in certain domains too. Hershcovich et al. [47] point out that these disparities in NLP occur across cultures as well (e.g.- resource gaps and impact between industrialized western cultures and developing world cultures). This is where Transfer Learning becomes useful. *Transductive* [48] and *inductive* transfer appear as two categories of Transfer Learning in the NLP literature where *inductive transfer* deemed to be more efficient and effective as a model can derive a hypothesis which works on unseen data samples as well. We expect to transfer already accumulated knowledge from a source task (or domain) to a target task (or domain) where the target task lack sufficient training data for the model to learn. In NLP, pre-trained models are expected to learn about a language at different levels (semantic, syntactic, and if possible at

pragmatic and discourse levels) at the pre-training stage where it is benefitted from large amounts of training data. When it is utilized for a target task, a much smaller dataset for the target task can be used to fine-tune the pre-train model while leveraging the already learned knowledge in the pre-training phase.
In multilingual models, Transfer Learning is expected to take place between languages as well. English is usually the preponderant language in multilingual pre-trained models which possess a major part in the pre-training corpora. It also has abundant resources for many target tasks. Interestingly, low-resource languages could be transferred knowledge from high-resource English. An example is zero-shot learning, originally proposed by Chang et al. [49] for text categorization and used in numerous use cases. Furthermore, it has been observed that a target task can obtain knowledge from different types of tasks too, as seen in multi-task learning scenarios [15]. There, a BERT model is fine-tuned with mini-batches of data points from each task while model parameters get updated according to the current task's objective function.
As Qiu et al. [30] mention, there are several factors that affect the transfer of knowledge. Model architecture, pre-trained data domain, the type of pre-training objective utilized by the model are some of them.

2.5.4 Continual Learning

Continual Learning (CL) refers to continuously learning over time while keeping the knowledge learned from previous stages [50]. In this way, models could learn from different data sources which may help the final target task by providing additional knowledge to the model through Transfer Learning. However, a pragmatic issue with this approach is *catastrophic forgetting* or *catastrophic interference*. This is a phenomenon that makes the model forget previously learned knowledge under the presence of a new data distribution. When adapting CL to downstream tasks such as sentiment classification, researchers have tried to address this issue. Ke et al. [51] proposes a capsule network-based solution for successful CL on an aspect-based sentiment classification task with BERT. Ke et al. [52] study about CL for sentiment classification, where they train a sequence of sentiment classification tasks and observe the effects of catastrophic forgetting.

2.6 Measuring the performance of the classification models

Accuracy, Precision, Recall and F-score are conventionally used metrics for measuring the performance of a classification model. A traditional definition of the number of correctly/incorrectly predicted samples in a binary classification scenario is;

- *True Positives (TP)* - Predicted as positive (or 1), true label is positive.

- *True Negatives (TN)* - Predicted as negative (or 0), true label is negative.
- *False Positive (FP)* - Predicted as 1, true label is 0.
- *False Negative (FN)* - Predicted as 0, true label is 1.

Then, the above mentioned performance metrics can be defined as below.

- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
- Precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{TP+FN}$
- F1 = $2 * \frac{Precision*Recall}{Precision+Recall}$

Although the accuracy metric provides a trivial way to account for the ratio of correct predictions against all the predictions, it could perform poorly when the dataset is unbalanced and hence cannot provide a correct measure of the model's performance. When F1-score is extended to a multiclass classification scenario, per-class F1-scores are calculated and then they are averaged to get a final F1-score. Based on the method of averaging, F1-score can also differ as macro, weighted, or micro. Macro F1-score can reflect the model's true prediction ability on each class without weighting following the class sizes which could provide more accurate results for the model's performance on a dataset.

CHAPTER 3

LITERATURE REVIEW

In this section, we present the related work present in contemporary literature. We divide this section into two subsections, with the first part describing work related to the first part of our experiments, which is evaluating and identifying best-performing models from monolingual and multilingual models which support the Sinhala language. In the last subsection, work related to the improvement of text classification in low-resource languages including the Sinhala language is presented.

3.1 Monolingual and multilingual pre-trained models

As mentioned previously in Chapter 2 also, a monolingual model is specialized for a single language whereas multilingual models typically support multiple languages. Mostly, it is a limited set of resource-rich languages that has monolingual Transformer-based pre-trained models built for them. The original BERT model (English), RoBERTa (English), FlauBERT (French; [53]), CamemBERT (French; [54]), ALBERTO (Italian; [55], Chinese BERT [56], AraBERT (Arabic; [57] are a few examples. There are few attempts at building such language models for resource moderate languages such as Finnish, Vietnamese, and Afrikaans [58–60] and even less so for low-resource languages, the reason being that pre-trained models require a substantial amount of text in its pre-training stage. Table 3.1 provides a glimpse of the sizes of pre-training corpora required to train such a model.

TABLE 3.1: SIZES OF CORPORA USED FOR PRE-TRAINING SEVERAL MONOLINGUAL LANGUAGE MODELS

Model	Corpora used	Size
BERT	BookCorpus, English Wikipedia	13GB/3.3B words
AraBERT	El-Khahir, OSIAN	24GB/70M sentences
FlauBERT	WMT-19, OPUS etc.	71GB
FinBERT	News, Discussion, WebCrawl	234M sentences

Note. Sizes are reported as they are mentioned in the respective original papers

Multilingual models are based on their monolingual counterparts where mBERT was the first of the kind. It is clear that multilingual models require multilingual corpora for pre-training and in most of them low-resource language representation would be minuscule. Table 3.2 provides parameters of a few of such multilingual models and the portion of Sinhala language in the pre-training datasets used to train them. It

TABLE 3.2: PORTION OF SINHALA INCLUDED IN MODEL PRE-TRAINING OF A FEW MULTILINGUAL MODELS

Model	Supported languages	Pre-training corpora	approx. % of Sinhala
LASER	93	EuroParl, United Nations, OpenSubtitles-2018, Tatoeba, Tanzil, Global Voices	3.6
LaBSE	109	CommonCrawl (v.2019-35), Wikipedia	0.4
XLM-R-base	100	CommonCrawl	0.15
XLM-R-large	100	CommonCrawl	0.15

is debatable whether the ideal type of model is monolingual or multilingual for any given downstream task. In the existing literature, conclusive evidence is not present to deem either of them superior to the other. Wu and Dredze [61] suggest that the possible reason for varying performance could be the quantity of language data used for pre-training the models. Nguyen and Tuan Nguyen [59], Le et al. [53] and Virtanen et al. [58] represent some of the work that compares monolingual model performance against that of multilingual models. PhoBERT is collectively referred to two BERT-based Vietnamese monolingual models pre-trained on a word-level 20GB (3 billion tokens/words) corpora consisting of Wikipedia and Vietnamese news text. This amount of data is remarkably low compared to the portion of Vietnamese text data included in CC-100 [11] (corpora used in XLM-R pre-training), which is 24.8 billion tokens. Nonetheless, PhoBERT-large model has outperformed the multilingual XLM-R model in NER, Part-of-Speech tagging (POS) , Dependency parsing and Natural Language Inference (NLI) tasks while also having a less amount of model parameters than XLM-R-large.

Finnish monolingual BERT (FinBERT) uses a pre-training corpus consisting of 234 million sentences extracted from different sources (news, online discussions, and web-crawled data). They (FinBERT) report superior results in several NLP tasks such as POS tagging, dependency parsing, NER, and a multiclass text classification task compared to multilingual mBERT. FlauBERT was built concurrently with similar models for the French language, such as CamemBERT [54] but uses a fewer amount of pre-training data. FlauBERT is presented in two variants; FlauBERT-small and FlauBERT-large. It utilizes a large pre-processed training corpus sized 71GB for pre-training the model and reports competitive results with CamemBERT and considerable gains (over 1%) to that of mBERT in tasks such as word sense disambiguation and dependency

parsing. For paraphrasing, both CamemBERT and FlauBERT perform slightly better than mBERT (below 1% gains) but for natural language inference on the French XNLI dataset, XLM-R-large model outperforms both of them by a substantial margin [53, 54].

The main constraint of monolingual models is that they are confined to individual languages. These models can be built only for languages that have an adequate amount of training data. Massively Multilingual Language Models (MMLMs) such as mt5, mBERT, and XLM-R, offer an alternative to overcome the difficulty of extending monolingual models to low-resource languages and means to support a large number of languages within one model. Yet, it is difficult to include desirably many languages in the model without the cost of having a large number of parameters, which is also known as *curse of multilinguality* [11]. Moreover, the representation of low-resource languages can still be low in multilingual models which results in a low performance compared to that of high-resource languages included in MMLMs. Hence, an auxiliary method to alleviate these obstacles is using multilingual models pre-trained only for a set of related languages. This argument can be backed by the reported performance drops when a language's representation is low in a multilingual model [61]. An example of a model built for a limited set of related languages is IndicBERT [62][1] which is an ALBERT [63] based model trained on IndicCorp; a corpus containing 452.8 million sentences belonging to 11 indic languages and English. IndicCorp, which contains text crawled from contemporary news websites, comprises 4 times more data for the indic languages, than that of corresponding OSCAR[2] and CC-100 which was used for pre-training XLM-R. IndicBERT, which also has *base* and *large* versions of it, shows better results for different tasks such as NER, multiple-choice question answering and other classification tasks. However, the performance gain is not consistent across all Indic languages where on some instances mBERT and/or XLM-R give better results. IndicBERT outperforms both mBERT and XLM-R by a considerable gap in some tasks such as question answering and cross-lingual sentence retrieval. This happens when the average result for a specific task across all the indic languages is considered. Furthermore, IndicBERT performs better than mBERT and XLM-R (on average) on the IndicGLUE benchmark [62]. However, for other additional tasks considered in their work such as sentiment classification, and genre classification on different datasets, XLM-R surpasses IndicBERT on average.

Bilingual models are another viable option which are pre-trained for a pair of languages. Usually, they are used for translation (or NLG) tasks that involve two languages. Khalid et al. [64] build a bilingual model by further pre-training an English BERT model for the Urdu language (in Latin script). However, they evaluate the model performance with performance metrics relevant to the pre-training stage. They observe

[1] https://indicnlp.ai4bharat.org/indic-bert/
[2] https://oscar-corpus.com/

that their bilingual model performs better compared to the counterpart monolingual model and mBERT which was further pre-trained on Urdu data, but lags behind the English BERT model.

Another possible way to incorporate languages that are low-resource and unsupported by contemporary language models is to use a language model trained for a language sharing similar characteristics. Ács et al. [65] show that fine-tuning on Russian monolingual model RuBERT [66] or English BERT (BERT) can provide satisfactory results through cross-lingual transfer on NER and POS tagging tasks on Uralic languages. They show that cross-lingual transfer happens regardless of the genetic similarity between Uralic languages and acceptable results can be achieved on datasets from unsupported Uralic languages as well. For morphological tasks, the monolingual model of the considered language gives the best result on par with XLM-R-base and mBERT. Yet, as seen previously as well, the performance of monolingual models does not consistently supersede that of multilingual models across all tasks or languages considered. XLM-R performs better than RuBERT for two unsupported Komi-Permyak and Komi-Zyryan languages which use the Cyrillic alphabet. FinBERT performs well for the unsupported Karelian language in POS tagging, but both XLM-R and mBERT outperform every monolingual model for the Livvi language in POS tagging. XLM-R also gives generally strong performance for all selected Uralic languages across all tasks. Their results also depict that the performance of multilingual models (XLM-R primarily) lies closer to that of monolingual models frequently, while substantially surpassing them on a few instances. They further suggest the large subword vocabulary of XLM-R as the reason for the orderly performance of XLM-R in their experiments.

The choice between multilingual models and monolingual models can depend on several factors. Monolingual models are favored over multilingual models mostly due to the fact that they do not suffer capacity dilution where the model capacity is shared among multiple languages [67]. Usually, the multilingual models are selected expecting to make advantage of cross-lingual knowledge transfer from high-resource languages to low-resource languages in downstream tasks, where insufficient annotated data is available for low-resource languages. However, different studies [61, 67–69] suggest that such knowledge transfer would depend on parameters such as the shared vocabulary, size of pre-training corpora, model parameters, and the alignment of representations across languages. On the other hand, Aguilar et al. [70] and Lauscher et al. [69] showed that multilingual pre-trained models' performances are not consistent across every NLP task. According to Aguilar et al. [70], these models are better at syntactic analysis as opposed to semantic analysis. Groenwold et al. [71] and Lauscher et al. [69] showed that the performance of a pre-trained model for a given language is heavily affected by the language family. In other words, if more related languages are included in the model, then it becomes beneficial for a language.

As a result, pre-trained models have been able to show better performances for Indo-European languages [68]. Some other researchers have experimented on different settings, such as zero-shot performance on languages that are included in the pre-trained model [36, 68, 72, 73], and performance on languages not included in the pre-trained models [74]. As an example, Fujinuma et al. [75] analyzes how multilingual model pre-training affects particularly unseen target languages. They show that including additional languages during pre-training improves the zero-shot performance for unseen target languages, when a model can be adapted for the target languages. This adaptation can be done either by using adapters [2], further training on MLM objective, or by extending the model's vocabulary.

3.2 Improving text classification results on Transformer based pre-trained language models

The rudimentary approach for text classification with Transformer models is fine-tuning an annotated dataset on an encoder model, by applying a task-specific header on top of the model (e.g. a feed-forward neural network for multi-class classification) as also mentioned in section 2.5. For improving text classification results, data augmentation has been proposed as an easily approachable method, which in turn helps to prevent overfitting of the model to low dataset sizes. Data augmentation can usually happen by making changes to the original data. Wei and Zou [13] propose a set of augmentation methods for text classification tasks. Synonym replacement, random swap, random insertion, and random deletion are the subtasks they suggest to be executed to augment a given dataset. Back translation is another method that can be followed to synthesize data points [76]. Bayer et al. [77] present a thorough survey and a taxonomy for data augmentation techniques for text classification.

Other than that, Sun et al. [37] propose a set of novel ways to fine-tune BERT for classification tasks. Further pre-training the model with in-domain, cross-domain or within-task data, multi-task fine-tuning, and vanilla fine-tuning for the target task are the main components of the proposed methods of fine-tuning. Vanilla fine-tuning then can be extended to picking out specific layers for fine-tuning and using different learning rates per layer. They combine these to improve the performance of several text classification tasks such as sentiment analysis. With Mutli-task learning one expects to transfer knowledge from other tasks to the desired target task. This could be beneficial when annotated data is insufficient for a particular target task or when we want to prevent the model from overfitting for a single task and becoming universal for a set of tasks [15]. Liu et al. [15] use a set of NLU tasks for multi-task learning which they name MT-DNN. Then a BERT model is used in which the encoder layers are shared among the tasks and task-specific headers are used for each task. Using an

algorithm that shuffles data points from different tasks, they fine-tune the pre-trained model and achieve improvements on several NLU benchmarks. However, this can be intricate in implementation and require enough datasets across multiple tasks, which might not be viable for low-resource languages. The likes of Meng et al. [78] propose innovative methods to improve text classification. They introduce a method for weakly-supervised text classification which uses a self-learning approach. Using class label names and related terms, they make the language model self-learn to perform classification. They experiment with their method on sentiment classification and topic classification benchmarks.

3.2.1 Text-classification of low-resource languages

As mentioned in the earlier sections, the paramount task in NLU for low-resource languages is to overcome the scarcity of resources such as corpora and datasets. However, the emergence of social media platforms has enabled a potential source to build new corpora or datasets for such languages. Efforts such as that of Muhammad et al. [79] and Kakwani et al. [62] are examples of building resources for low-resource languages. However, these improvements are still scant compared to the dominance of high-resource languages in NLP. As an example Aggarwal et al. [80] show that resource availability directly affects the downstream task performance for NLI tasks in Indic languages, where mid-resource Hindi and Bengali perform well while Odia suffers low performance. On the other hand, low-resource languages such as Marathi, and Kannada perform fairly well due to their script-sharing higher-resource language siblings Hindi and Tamil (respectively).

It was mentioned in Chapter 2 that it is viable to utilize MMLMs to perform NLP tasks involving low-resource languages than building monolingual models from scratch. The performance still lags for resource-poor languages even with this procedure. One approach to improve an MMLM's performance is enhancing the alignments between resource-rich and resource-poor languages' representations in the model's vector space (typically at the word level). The intention of this is to facilitate knowledge transfer from high to low resource language better. Zero-shot learning is a popular approach in this scenario. Müller et al. [81] explore how a multilingual model facilitates cross-lingual knowledge transfer in a zero-shot scenario. They use an analyzing technique which randomly initializes specific layers of an mBERT model. Then the modified model is fine-tuned and compared against another fine-tuned model which was initialized from the pre-trained weights, which is followed by applying a similarity metric to measure similarities of representations across languages. They show that cross-lingual representations and their alignments are learned at the lower layers while the upper layers are language agnostic. These alignments are present since the

pre-training stage as they report. On top of that, cross-lingual alignment directly impacts the cross-lingual transfer. However, they experiment with word-level tasks such as POS, NER and report their findings mismatch with the work of Singh et al. [82] which conducts similar experiments with NLI, which is not a word-level task.

3.2.1.1 Cross-lingual alignment of models

It is also suggested that an explicit alignment objective should be followed to obtain improvements in Transformer models such as BERT which have not been explicitly trained on such objective for alignment [83]. Linear mapping [84], contrastive learning [85] based objectives are example techniques for alignment. Wu and Dredze [83] further compare among a few alignment methods using mBERT and XLM-R for tasks such as Named Entity Recognition (NER) . However, these alignment methods require high-quality parallel text (bi-text) such as Europarl [86], which maps English words/entities with other language words. Wu and Dredze [83] suggests that increasing model capacity as seen in XLM-R models can lead to obtaining better cross-lingual representations without an explicit alignment objective. Yet, these methods do not perform consistently and tend to produce degraded results with noisy bi-text, when tested on zero-shot learning tasks which are useful when the target language's fine-tuning data is insufficient for the given task (i.e.- The alignment objective is carried out for a target language with the use of bi-text. Then, the target task for the target language is carried out by zero-shot learning). Singh et al. [82] claim that the vector representations in mBERT model do not reside in a shared space as expected. Rather, mBERT creates partitions in spaces to reflect linguistic and evolutionary associations between languages. Additionally, they also claim that the tokenization (word level, character level, subword etc.) used affects the forming of this vector space. This is an important finding as it has a direct impact on utilizing Transfer Learning for low-resource languages in multilingual models.

Efimov et al. [87] analyze how cross-lingual adjustment of multilingual models such as mBERT affect on zero-shot learning results. They first cross-lingually adjust an mBERT model using parallel data for English paired with 4 other typologically different languages followed by a fine-tuning of English data for NLI, NER, and question answering. They observe that the expected performance increase does not occur for all the tasks and languages (question answering shows no significant improvements and shows degradation sometimes. Similar phenomena for some languages such as Spanish) and the amount of parallel data only benefits the NER, NLI tasks. Furthermore, they show the distances between related and unrelated words are merely affected and for a task like NER, and when the model is only fine-tuned (without adjustment) for the target task with English data, both related and unrelated words can be grouped closer

after the adjustment process.

Earlier, Liu et al. [88] had shown that vanilla fine-tuning of a multilingual model for a target task could degrade its cross-lingual abilities and hinder the model's performance on its original task (e.g.- mBERT's original task is MLM). As the proposed solution, they utilize *Gradient Episodic Memory* [89] framework to constrain the fine-tuning process. They also show that their method can achieve better performance in zero-shot tasks for POS and NER after experimenting with mBERT and XLM-R.

3.2.1.2 Lexicons as external knowledge bases

Lexicons can be identified as a collection of words (or vocabulary). Sentiment lexicons contain sentiment words which are labeled with their sentiment values such as *positive (1), negative (-1), neutral (0)* [90] or with more fine-grained labels with continuous values [91–93]. Rather than recording polarity scores for sentiment values of words, some lexicons present numerical values of different measures such as valence score [91]. In this sense, lexicons can be considered as external knowledge bases as well which can be used to provide the model with additional factual or linguistic knowledge.

In early attempts, lexicons have been used as an unsupervised way for sentiment classification based on the aggregated sentiment score given by the words composing a sentence or a document [94]. Musto et al. [95] also present sentiment classification methods based on sentiment scores for microblog posts. Lexicons have also been used with different Deep Learning paradigms. Shin et al. [96] used lexicons with CNNs. Lexicons have also been used with RNNs [97–99]. More recently, lexicons have been used as knowledge bases with Transformer models as well. Suresh and Ong [100] used sentence vectors created using lexicons as an additional input to the BERT model for emotion classification. Ke et al. [101] and Zhong et al. [102] also present attempts of incorporating lexicons as external knowledge bases with Transformer models. An obstacle which rise when using lexicons is that lexicons of sufficient quality are mostly limited to high-resource languages such as English. For languages like Sinhala, such lexicons are rarely published and very much limited in number.

3.2.2 Sinhala Text Classification

Sinhala language is ailed by the scarcity of resources and the amount of research carried out also lacks [103]. There are scattered attempts of developing resources (corpora and language tools) for the Sinhala language. Some such instances are a POS tagger tool for Sinhala by Fernando et al. [104], NER system for Sinhala by Manamini et al. [105], morphological analyzer by Kumarasinghe et al. [106], Upeksha et al. [107]'s attempt on building a corpus for Sinhala language, a similarity measure based approach

for clustering Sinhala news documents [108] and a Sinhala lexicon built by Weerasinghe et al. [109]. For NLG applications such as machine translation, mBART has shown some fairly decent performance [110].

Early research in Sinhala text classification has been mainly limited to classical approaches. Experiments with traditional machine learning methods such as Support Vector Machines (SVM) were carried out by [111]. Furthermore, approaches such as rule-based systems [112], a stop word extraction method for text classification using TF-IDF [113], Feed-forward Neural network based system [114] and a Word2Vec based approach[3] have also been followed. Chathuranga et al. [115] proposed a method for Sinhala text classification based on a lexicon. Jayasuriya et al. [116] conduct sentiment analysis on Sinhala social media data on the sports domain. Ranathunga and Liyanage [117] is the first to experiment with Deep Learning techniques such as Convolutional Neural Networks (CNN) and LSTM networks-based techniques for Sinhala sentiment classification tasks. Moreover, they have built Word2Vec and fastText word embedding models for the Sinhala language. However, these methods are not benefitted from contextual embeddings produced by Transformer based language models. Demotte et al. [118] also proposed an LSTM-based system for Sinhala text classification based on S-LSTMs [119]. Senevirathne et al. [120] empirically analyzed RNN, Bi-LSTM, and Capsule Networks for Sinhala news text sentiment classification. SinBERTo and Sinhala-RoBERTa are two pre-trained RoBERTa-based MonoLMs for Sinhala, that have been released recently. To the best of our knowledge, these models do not have related research work published, nor have been used in text classification [121] by the time of the publication of this work.

[3] http://bit.ly/2QKI9Np

CHAPTER 4

EVALUATING TRANSFORMER-BASED ENCODER MODELS

In this chapter, we discuss the first main task of this research, which is the evaluation of existing monolingual and multilingual Transformer encoder-based language models on Sinhala text classification performance. We use Sinhala text classification datasets spanning four different domains. We also introduce some of them as new datasets which can be used in potential future research work.

4.1 Task description

As discussed in Chapter 3, it is necessary to empirically identify the best-performing models for text classification in the Sinhala language. This is since (as discussed in 3.1) there is no clear winner between pre-trained multilingual and monolingual models. Such a model also cannot be selected only by the results shown for other languages/datasets. Hence, we selected available pre-trained Transformer based encoder language models (XLM-R, LaBSE, and RoBERTa based Sinhala monolingual models) and non-Transformer based language models (LASER), which are either monolingual or multilingual. Major details about some of the selected pre-trained models are discussed in section 2.4. Additionally, we pre-train new SinBERT Sinhala monolingual models with a large Sinhala corpus ("sin-cc-15M") and add it to our series of experiments for text classification. We then empirically obtain the performance of each model on 4 different classification tasks in Sinhala and select a consistently best-performing model from the selected models, on all the tasks.

4.2 SinBERT models

We build a new set of Sinhala pre-trained monolingual encoder models which we dub SinBERT. RoBERTa has shown improved results over other competitive models for the GLUE benchmark, specifically for classification tasks. Hence, we build the SinBERT models based on RoBERTa. We use Huggingface's Transformers libraries in Pytorch to pre-train our RoBERTa models[1]. AdamW [122] is used as the optimizer with the hyperparameters; a batch size of 16, a learning rate of 1e-4, and a maximum of two training epochs to pre-train the models. We introduce two variants of our model, namely; SinBERT-small which has 6 hidden layers, and SinBERT-large containing 12 hidden layers. Parameters of the two models are shown in Table 4.1.
SinBERT models are pre-trained using the "sin-cc-15M"[2] corpus. Currently, it is the

[1]We publicly release the pre-training and fine-tuning codes on https://github.com/nlpcuom/Sinhala-text-classification

[2]shorturl.at/qAUV1

largest monolingual corpus available for the Sinhala language to the best of our knowledge. The dataset has 15.7 million sentences which were extracted from 3 sources: OSCAR, CC-100, and raw text data from Sinhala news websites. CC-100 dataset contains 3.7GB of data for Sinhala and OSCAR contains 802MB of Sinhala text including duplicated text. The raw news data extracted from Sinhala news sites is 413MB in size. The final sin-cc-15M dataset has been cleaned of other language words/characters and invalid characters. Cleaned dataset statistics are shown in Table 4.2.

TABLE 4.1: PARAMETERS OF THE SINBERT MODELS

Parameter	SinBERT-small	SinBERT-large
Hidden layers	6	12
Attention heads	6	12
Max. Position embeddings	514	514
Vocabulary size	30000	52000
Number of trainable parameters	66.5M	125.9M

TABLE 4.2: STATISTICS OF THE PRE-TRAINING CORPUS

Number of words	192.6M
Number of unique words	2.7M
Number of sentences	15.7M
Average number of words/sentence	12.2

4.3 Related work

4.3.1 Other existing Sinhala monoligual models

Only a few Sinhala monolingual models have been trained and publicly released. Almost all of them have not been benchmarked or included in any published work to the best of our knowledge. They have been pre-trained using smaller corpora compared to the sin-cc-15M dataset. We select a couple of available Sinhala monolingual models on Huggingface, which are also RoBERTa based. Namely, they are SinhalaBERTo and SinBERTo which have 52000 size vocabularies. SinBERTo has been pre-trained on a small news corpus whereas SinhalaBERTo has been pre-trained on a larger Sinhala OSCAR dataset. The two models also share a similar model architecture. Sinhala-Roberta-Oscar[3] and sinhala-roberta-mc4[4] are another pair of Sinhala monolingual models available publicly, by they have smaller vocabularies. Hence, we could claim that our SinBERT models have the advantage of a much larger pre-training corpus compared to the existing models.

[3]https://huggingface.co/keshan/sinhala-roberta-oscar
[4]https://huggingface.co/keshan/sinhala-roberta-mc4

4.4 Datasets and sub-tasks

The set of classification tasks in line with the above description is explained below.

4.4.1 Sentiment analysis

Sentiment analysis concerns classifying the sentiment value of a text at the basic level, which can be further developed into tasks such as Aspect Based Sentiment Analysis (ABSA) and emotion recognition which are more fine-grained classification tasks. In our experiments, we consider the classification of Sinhala document-level text (which may consist of multiple sentences) into 4 sentiment levels, namely *Positive, Negative, Neutral* and *Conflict* using the dataset published by Senevirathne et al. [120]. The dataset constitutes 15059 Sinhala news comments extracted from Lankadeepa[5] and GossipLanka[6] and annotated by human annotators with a Cohen's Kappa value of 0.65 for the inter-annotator agreement score. The average document length is 21.66 words of the dataset. Table 4.3 details the dataset statistics. A weighted F1-score-based baseline is reported by Senevirathne et al. [120] using Capsule Networks and recurrent neural networks.

4.4.2 News category classification

We use a dataset introduced by de Silva [123], which includes news text spanning across 5 different domains namely *Political, Business, Entertainment, Science* and *Technology*. The aim is to classify the news text belonging to different domains of news. We pre-process the dataset and exclude sentences which are shorter than 3 words, such as celebrity names, place names. The total dataset contains 3327 sentences with an average length of 23.49 words per sentence.

4.4.3 News source classification

Here, we use a newly compiled dataset that aims on classifying Sinhala news headlines according to their source. The news source dataset comprises headlines that have been web crawled from 9 different sources (Sinhala news websites) which are mentioned below. We reduce the size of data in the original web-scraped dataset [124] with the purpose of handling the contrasting class imbalance. Furthermore, we exclude one news source (Sinhala Wikipedia) from the originally scraped dataset as it mostly contains invalid characters, numbers and single word sentences. The final dataset has sentences with a 8.42 average word length a total of 24093 sentences.

[5]https://www.lankadeepa.lk/
[6]https://www.gossiplankanews.com/

- Sri Lanka Army - https://www.army.lk/
- Dinamina - http://www.dinamina.lk/
- GossipLanka - https://www.gossiplankanews.com/
- Hiru - https://www.hirunews.lk/
- ITN - https://www.itnnews.lk/
- Lankapuwath - http://sinhala.lankapuvath.lk/,
- NewsLK - https://www.news.lk/
- Newsfirst - https://www.newsfirst.lk/
- World Socialist Web Site-Sinhala - https://www.wsws.org/si.

4.4.3.1 Writing style classification

We extracted text from Upeksha et al. [125]'s large Sinhala corpus, which contains text spanning across a set of different genres. However, their publicly available corpora do not contain the complete amount of data mentioned in their paper. In this task, we classify documents based on their genre/writing style. We select text under 4 categories, which are *News, Academic, Blog, Creative*. We pre-process the extracted text by deduplicating, removing English-only text and removing very long documents (lengths larger than 3500 characters). Since the dataset contains very long text, we use truncation to fit them into the models. No prior evaluation (baseline) has been presented for this dataset. The final dataset has an average length of 181.97 per document and a total amount of 12514 documents for the 4 different writing styles.

TABLE 4.3: STATISTICS OF THE 4 CLASSIFICATION DATASETS

Dataset/task	Max.data points of a class	Min.data points of a class	Total
Sentiment	7665	1911	15059
News source	3109	1541	24093
News categories	1019	438	3327
Writing style	4463	2111	12514

4.5 Methodology

In line with section 2.4, we select both monolingual (SinhalaBERTo, SinBERTo, SinBERT) and multilingual (XLM-R, LASER, LaBSE) models to test against the 4 different downstream tasks. Then, we fine-tune the selected models in the standard method

(vanilla fine-tuning) and obtain the results for the selected set of text classification tasks. Considering the macro F1-score (section 2.6), the best-performing model is determined. Table 4.4 presents the fine-tuning hyper-parameters. We use macro F1-scores as the primary metric for measuring performance in our experiments as they average all the classes with an equal weight regardless of the class size.

In the reported results, macro-averaged F1-scores are reported using 5 different randomly-initialized runs for each experiment using 4:1 train/test splits of the datasets, in order to deter the randomness of results. We use only 3 randomly initialized runs for LASER and LaBSE as their performance is much worse than that of XLM-R and the other monolingual models. For all the implementations, we use TFHub (with Tensorflow), Pytorch, and Huggingface which provides easy-to-use libraries for implementing Transformer models. In addition to the results with the full fine-tuning datasets, we conduct a set of experiments by varying the size of fine-tuning datasets. There, we compare the performance of SinBERT models with the XLM-R-base model.

TABLE 4.4: HYPERPARAMETERS FOR THE EXPERIMENTS FOR EVALUATING LANGUAGE MODELS

Hyperparameter	XLM-R	SinBERT	Other
Learning rate	5e-6	1e-5	1e-5, 5e-5
Batch size	16, 8	16	16
Epochs	5	10	5
Optimizer	AdamW	AdamW	AdamW

The completion time for each task depends on the size of the dataset given the batch size is constant (16 or 8).

4.5.1 Fine-tuning method used in our work

For the classification tasks similar to which are under the spotlight in our work, it is common to use the method shown in figure 4.1 to obtain the classification predictions from the model. There, [CLS] token's vector representation is utilized as the representation for a sentence/document. This is obtained at the final layer of the encoder. As an example, for XLM-R this would be a vector with a dimension of 768. For our LaBSE implementation which is purely based on Tensorflow[7] and TFHub[8], this is called "pooled representation". This representation is then fed into a feed-forward neural network. For the LASER model, (which is not a Transformer-based mode) fixed vector representations which have a size of 1024 for the sentences/documents, are sent through a linear layer for a dimensionality reduction. In the figure 4.1 the [EOS] token is another special token added to the text to resemble the end of the sentence/text. The

[7]https://www.tensorflow.org/
[8]https://tfhub.dev/

other tokens (S_i) represent the actual tokenized words or subwords of the input text. The output from the classifier head (logit vectors) is then sent through a final softmax layer, which yields a probability distribution for the predictions for each class given an input text. The softmax layer could be integrated into the classifier head. Cross-entropy loss is used to calculate the loss function as we work on multiclass classification scenarios. Then, the final prediction can be inferred as the class with the highest probability assigned by the softmax operation (Eq 4.1).

We used the vanilla fine-tuning process, where the [CLS] token output from the encoder part of the pre-trained model is fed to a feed-forward neural network-based classifier. For Sinhala monolingual models, we use Huggingface's default classifier for RoBERTa models. A linear layer preceded by a dropout layer was used as the classifier head for LASER and LaBSE. For XLM-R-large, we use 8 as the batch size due to hardware (GPU) limitations. All the training tasks (pre-training, fine-tuning) were conducted on an NVIDIA Quadro RTX 6000 (24GB) GPU.

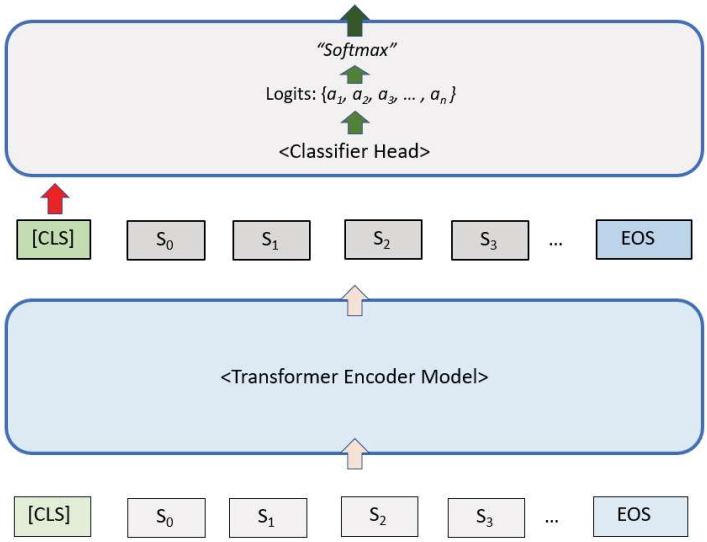

Fig. 4.1: Utilizing the CLS token representation

$$prediction = \underset{i}{\operatorname{argmax}}(softmax(\{a_i\}_{i=1,...,n})) \tag{4.1}$$

The classifier head consists of several sub-layers as shown in figure 4.2 below. The implementation of classifier head for Roberta models with Pytorch [126][9] (also for XLM-R) by Huggingface[10] is shown here. There the *num_labels* refers to the number of classification classes used in the dataset.

[9]https://pytorch.org
[10]https://github.com/huggingface/Transformers

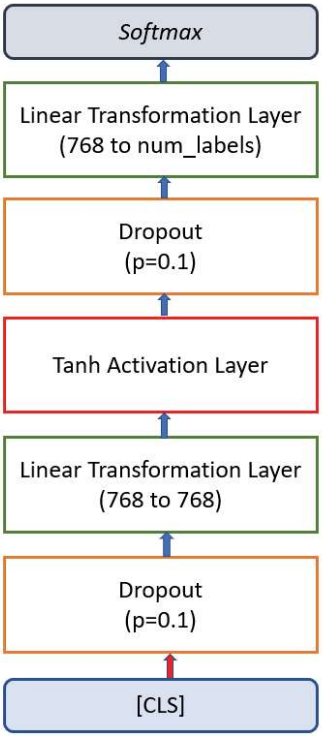

Fig. 4.2: Classifier head composition for Transformer models

4.6 Results

We report the macro-F1 score for each task using the selected models in Table 4.5. We also report some available, previous baseline results for the classification tasks as well. XLM-R models (base and large) consistently produce better results for the classification tasks. XLM-R-large turns out to be the best model among the monolingual and multilingual models. For the sentiment analysis task, Senevirathne et al. [120] report the baseline results in weighted-F1 score which we have indicated in Table 4.5. We do not utilize LaBSE for all the tasks as it performs very poorly on the initially tested tasks. The LASER model does not perform consistently for all the tasks although it shows a considerable performance for the writing style classification task and a comparable performance for the sentiment analysis task. Even though the newly pre-trained SinBERT models are on par with XLM-R models (fall behind XLM-R-large), they are consistently ahead of the other monolingual models that we experimented with.
Figures 4.3-4.6 show the pattern of macro-F1 scores obtained by changing the fine-tuning dataset sizes for each classification task. Here, we use the SinBERT models and the XLM-R-base model for comparison. The selection of the XLM-R-base is mainly due to computational resource constraints. We can clearly observe that for low dataset sizes, SinBERT models yield better results than the XLM-R-base. As the dataset size grows up, the XLM-R-base model gains performance and surpasses or stays on par

with the SinBERT models. This is except for the news source classification task, where SinBERT models stay dominant throughout all the dataset sizes. The reason for this could be that the raw corpora which the news source dataset was created from, has been used as a part of the pre-training corpus of the SinBERT models.

TABLE 4.5: RESULTS FOR THE MODEL EVALUATION ON THE FOUR CLASSIFICATION TASKS

Model	Sentiment	News sources	News categories	Writing style
Baseline	$59.42_{w.F1}$	-	-	-
LaBSE	20.63	11.85	24.09	-
LASER	54.07	28.84	48.54	87.06
XLM-R_{base}	58.08	58.29	85.12	96.89
XLM-R_{large}	**60.45** ($68.1_{w.F1}$)	**61.84**	**89.54**	**98.41**
SinBERTo	50.83	57.22	78.07	93.84
SinhalaBERTo	49.71	57.34	82.73	94.10
$SinBERT_{small}$	53.85	60.42	84.75	95.00
$SinBERT_{large}$	54.08	60.51	85.19	95.49

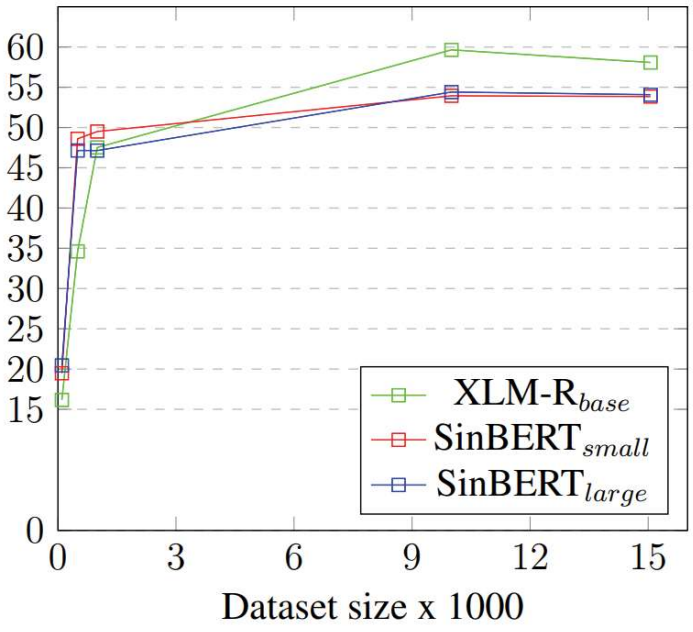

Fig. 4.3: macro-F1 for sentiment classification task with varying dataset size using SinBERT and XLM-R-base

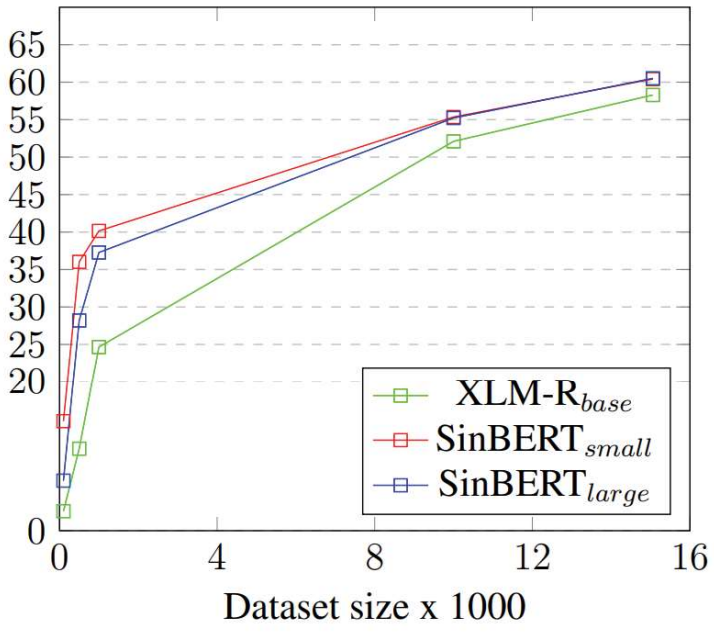

Fig. 4.4: macro-F1 for news source classification task with varying dataset size using SinBERT and XLM-R-base

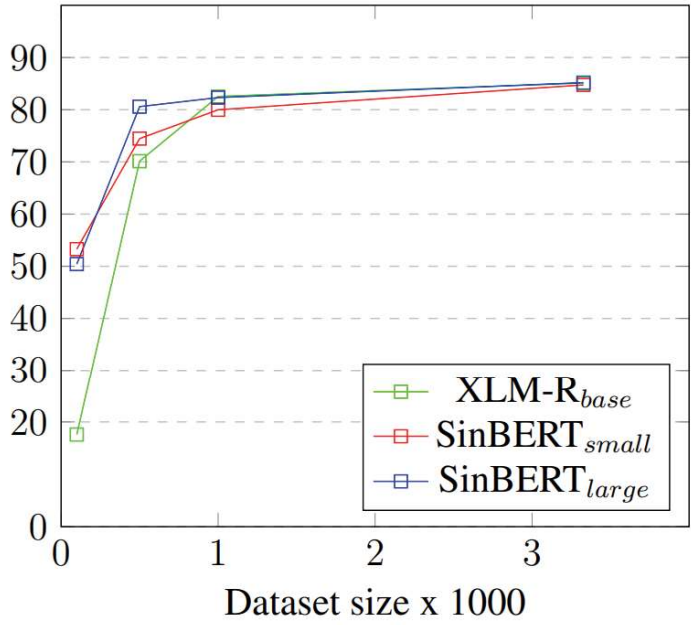

Fig. 4.5: macro-F1 for news category classification task with varying dataset size using SinBERT and XLM-R-base

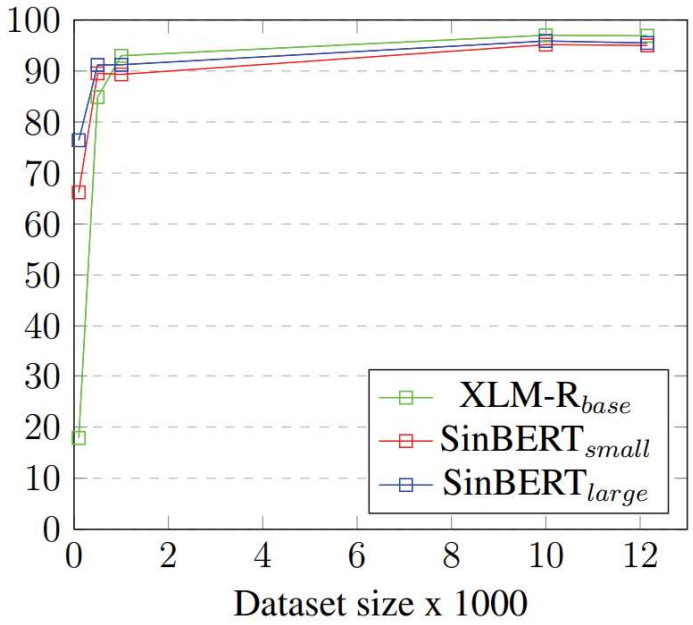

Fig. 4.6: macro-F1 for writing style classification task with varying dataset size using SinBERT and XLM-R-base

4.7 Discussion

As observed in the previous section, SinBERT models and XLM-R turned out to be the best models for the classification tasks. While the XLM-R-large model performs best across all the tasks, SinBERT models perform best among the other Sinhala monolingual models which shows that SinBERT could be the best option for text classification tasks when it comes to monolingual models. When compared with XLM-R-base, SinBERT models show better results when the fine-tuning dataset sizes are low. Hence, SinBERT models might be a good option for Sinhala text classification tasks with low annotated datasets. Out of the two SinBERT models, SinBERT-large does not offer a significant improvement in performance compared to the SinBERT-small. Hence, using SinBERT-small would not result in a significant trade-off in task performance against the computational resources used. Furthermore, this series of experiments shows that multilingual models can perform better for a classification task in a low-resource language, while a monolingual model pre-trained with sufficient amounts of corpora can also perform considerably well.

CHAPTER 5

IMPROVING ENCODER-BASED LANGUAGE MODELS FOR SINHALA TEXT CLASSIFICATION

Work carried out for improving the text classification results for the Sinhala language is discussed in this chapter, continuing from evaluating the pre-trained Transformer models for Sinhala text classification in the previous chapter. For convenience, the Sinhala sentiment analysis task is considered here. We experiment with a set of existing methods in this regard and propose a novel method that yields improvements in the sentiment classification task. We test the robustness of this method on a few other languages as well. Furthermore, we present some interesting observations regarding the proposed method and the selected languages.

5.1 Task Description

As mentioned in Section 3, out of the categories of sentiment classification, we focus on multi-class (4-class and 3-class) (or coarse-grained) sentiment classification. There, pre-trained models are fine-tuned for the downstream sentiment analysis task for a selected language dataset [120]. We utilize the multilingual XLM-R model, which was identified to be the best performing model for Sinhala text classification tasks from previous the chapter.

5.2 Alternative methods

Before arriving with a novel improvement for the sentiment classification task, some of the existing techniques were experimented with, on the Sinhala sentiment classification dataset. We provide concise descriptions about them and results produced by them in the following sub-sections.

5.2.1 Translation of input text from Sinhala to English

Training the model with translated data is commonly used as a baseline result for NLP tasks [7, 18]. There, typically the low-resource language data points are translated to resource-rich language and the model is trained (or fine-tuned) with the translated data. Under this method, we conduct micro experiments with minor modifications to the original idea. Which are, using different portions of the original dataset in Sinhala translated to English. A closely related approach is converting the text from target language script into to the Latin script, which produces a transliterated dataset. In our

experiment, we do not utilize that method as it is a time-consuming task to manually convert the datasets having over ten thousand data points.

5.2.2 Instance weighting

In line with Section 2.5, instance weighting is intended as a method which allows the model to put more weight on data points which are not well comprehended by the model. We modify the loss function such that more weight is added to the back-propagated loss when the calculated loss is higher for a particular data input. Equation 5.1 shows the update of model parameters θ with pre-calculated weights w_i. Equation 5.2, 5.3 shows how we calculate the weights. There, values of ϵ (smoothing constant) and τ (normalization constant) can be manually set. When ϵ is large, w gets closer to 1 and τ parameter makes sure that average of all weights from a mini-batch is 1. In Eq. 5.3 the weight is calculated using the standard deviation (std) of historical losses. h^{t-1} denotes the number of stored historical losses. Then we also mix these two types of weight calculations during the fine-tuning. Equation 5.4 shows mixing of the weights with the help of a variable β ($\in (0, 1)$) which again can be manually set.

$$\theta \leftarrow \theta - \alpha \sum_{i=1}^{k} w_i \nabla_\theta f(y_i, g_\theta(x_i)) \tag{5.1}$$

$$w = (-\mathbf{y}^T log\ p(\mathbf{y}|\mathbf{x}) + \epsilon)/\tau \tag{5.2}$$

$$w = (std(h^{t-1}) + \epsilon)/\tau \tag{5.3}$$

$$w = (\beta(-\mathbf{y}^T log\ p(\mathbf{y}|\mathbf{x}) + (1-\beta)(std(h^{t-1})) + \epsilon)/\tau \tag{5.4}$$

5.2.3 Using different prediction heads

Typically, the prediction head attached to the pre-trained model's encoder is a simple feed-forward neural network. However, one can add an intricate architecture as the classifier (prediction) head [127]. The generic classifier head architectures used in the publicly available Transformer libraries (such as Huggingface) are similar to the one shown in Figure 4.2. In our experiments, we acquire the results using a different activation function (ReLu) from the default activation function (GeLu) and also use an additional hidden LSTM layer in the classifier head.

5.2.4 Sequential Multi-task learning

As explained in Chapter 3, we can utilize multiple datasets from different tasks/domains to improve the classification performance of the model [15]. Multi-task learning can be a sequentially trained task or simultaneous training of the model with related tasks/datasets. The latter is harder to implement as the training phase should be designed to account

inputs from different data sources. With our datasets, we try the sequential fine-tuning of datasets where we first fine-tune the XLM-R-base model with high-resource English language sentiment classification data and then with the Sinhala sentiment analysis dataset. We also try the approach of fine-tuning the XLM-R-base model first with news domain classification data (Chapter 4) followed by the Sinhala sentiment classification dataset. There we use different classification head for the two datasets.

5.2.5 Data augmentation

There exist several techniques for data augmentations such as those of Wei and Zou [13], Feng et al. [128]. We also try a couple of data augmentation methods which we devise on our own. First one is replacing sentiment words using an external lexicon. There, we first identify sentiment words in a sentence using the lexicon and replace the words by translating them to English. Another technique that we try is using auxiliary labels. The concept of auxiliary labels is to hint the model of what sentiment a sentence would carry. Auxiliary labels are applied to the training dataset and during the inference time, test dataset is fed to the model without auxiliary labels attached to the test sentences.

5.2.6 Adapter based methods

Adapters were initially proposed as a parameter-efficient alternative for adapting language models for downstream tasks rather than vanilla fine-tuning the models or pre-training from scratch. Houlsby et al. [41] were among the first who proposed the technique of using adapters whereas Pfeiffer et al. [2] introduced modified versions of adapters and off-the-shelf methods to train and use adapters for Transformer based language models. Figure 5.1 shows the language and task adaptor architectures proposed by Pfeiffer et al. [2]. The MAD-X framework offers a mechanism to incorporate adapters to facilitate the adaptation of MMLMs for low-resource languages. Their language adapters are trained with unlabeled language datasets and the task adapters can be trained using a relevant labeled task-specific dataset.

5.2.7 Ensembling model weights

In this method, following the findings of [129] we add weights of two separately fine-tuned XLM-R-base models (for English and Sinhala) using a simple linear combination of the weights. As shown in Eq. 5.5, we control this addition with the help of a parameter α which takes values in the range $[0, 1]$. There, w_f represents the final model weights which is a linear combination of the weights of a fine-tuned XLM-R-base model on English (w_{en} and a Sinhala (w_{si}) datasets. Table 5.1 shows results for

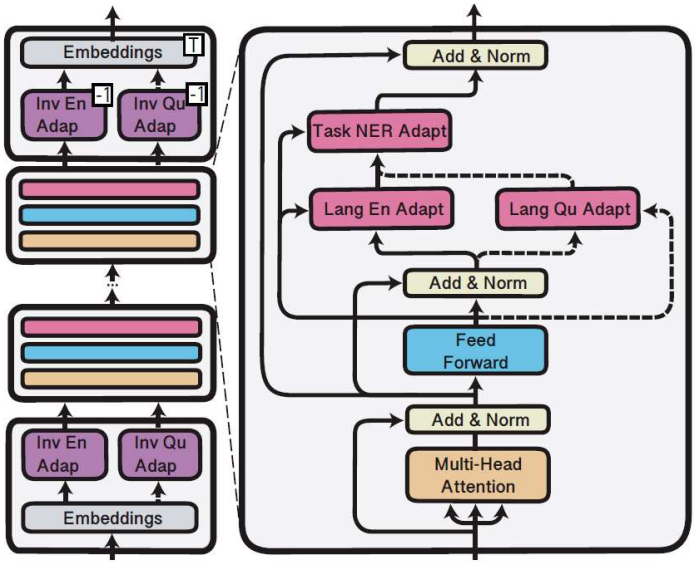

Fig. 5.1: Adapters introduced with the MAD-X adapter framework [2]

different values for the α parameter. The results were obtained by predicting the labels for the Sinhala sentiment classification dataset (test set).

$$w_f = \alpha * w_{en} + (1 - \alpha) * w_{si} \tag{5.5}$$

TABLE 5.1: RESULTS OBTAINED FROM THE WEIGHT ENSEMBLING EXPERIMENT

α	Accuracy	Precision	Recall	macro-F1
0	77.49	70.82	**69.23**	**69.89**
0.1	77.64	71.02	68.68	69.56
0.2	77.79	71.25	67.85	68.94
0.3	**78.02**	**72.13**	67.43	68.79
0.4	77.79	72.14	67.33	68.88
0.5	77.03	71.14	67.28	68.88
0.6	75.89	70.71	67.56	68.87
0.7	74.45	69.46	65.28	66.85
0.8	74.68	70.53	64.52	66.65
0.9	74.98	71.06	63.95	66.43
1	73.76	70.26	61.76	64.49

5.2.8 Injection of external vector embeddings

This method is similar to what we would discuss in section 5.3, which is incorporating external knowledge bases. The idea is to add additional external knowledge to the model during the fine-tuning phase. As depicted in figure 5.2 we concatenate external embedding vectors with the ones produced by the XLM-R model before feeding the aggregate to the classifier head. These embeddings are created using an external lexicon such as Mohammad [91]'s lexicon or by obtaining embeddings from a model such as LASER. At a more advanced level, we also try to adopt the method proposed by Suresh and Ong [100] to inject external vector embeddings. They utilize matrix operations to combine external vector embeddings with the ones from the model.
There we first create feature vectors of sentences by assigning the valence, arousal, and dominance scores (v, a, d) to each word forming a vector representation of the sentence. The output embedding for an n-word sentence is then represented by H_e as shown in Eq. 5.6 after being padded to have a fixed length (N). It is then combined with the embedding matrix (H_c) produced by the XLM-R model at the final layer of its encoder. If the [CLS] token representation (i.e- the first row of the matrix H_c) is denoted by h_0, then our final representation (h_l) is taken as shown in Eq. 5.8 and 5.9, which is fed to the classification head.

$$H_e = \{v_1 a_1 d_1 \; v_2 a_2 d_2 \; ... \; v_n a_n d_n \; x_1 \; ... \; x_{N-n}\} \tag{5.6}$$

$$K = concat(H_c, H_e) \tag{5.7}$$

$$s = softmax(h_0^T.K) \tag{5.8}$$

$$h_l = s^T.K \tag{5.9}$$

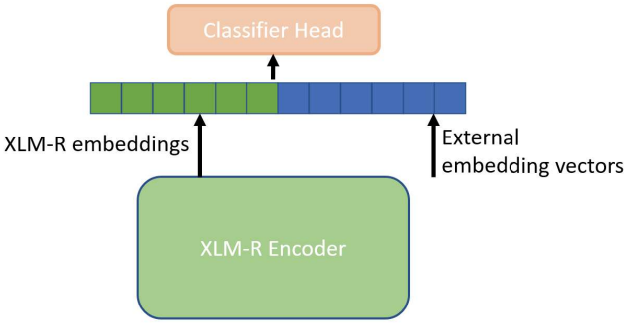

Fig. 5.2: Injecting external embedding vectors

5.2.9 Initial results

In Table 5.2 we present the results we obtained for the set of techniques we experimented with on the Sinhala sentiment classification dataset. Most of them failed to produce a superior performance compared to the baseline and many of them were falling behind the baseline result. Some of the techniques were performed with the initial 4-class sentiment dataset before adapting the 3-class version of the dataset. Hence, those results are also included in the table. All the results are reported as macro-F1 scores which were averaged across 3 randomly initiated runs.

TABLE 5.2: INITIAL RESULTS OBTAINED USING EACH ALTERNATIVE METHOD

Method	macro-F1 score
Baseline (*4-class*)	**63.48**
Multi-task learning (sequential-*with Si news domain dataset*)	60.88
Instance weighting	62.18
Different classifier heads (LSTM)	61.6
Replacing sentiment words with English translations	49.86
Baseline (*3-class*)	69.61
Multi task learning (sequential-*with En sentiment dataset*)	**70.98**
Sentences translated to English	26.89
Some sentences translated to English	68.65
Integrating external knowledge base (lexicon 5.2.8)	16.83
Further fine-tuning after integrating external lexicon (*from previous*)	59.37
Integrating LASER embeddings	61.37
Fine-tuning with adapters included (task adapter trained with En sentiment dataset)	70.45

5.3 Proposed Technique

After the initial experiments with the above set of techniques, we devise a novel method to improve text classification results of the sentiment classification task for the Sinhala language. Our method follows two steps of fine-tuning with one intermediate fine-tuning. This newly suggested method improves sentiment classification results for Sinhala and a few other languages. This proposed novel method can also be considered comparable to Ke et al. [101]'s idea of integrating external knowledge. However, we make use of lexicons instead of the pre-training technique used by them.

5.3.1 Motivation and related work

As mentioned in Chapter 3, existing techniques for low-resource languages mainly focus on explicit alignment of words or parameter-efficient methods such as adapters other than building additional language resources (i.e.- new corpora or data augmentation). Using parameter efficient methods does not offer superior results to standard

fine-tuning sometimes. The issue with most explicit alignment methods is that they are not trivial to implement. Inspired by the idea of auxiliary sentences proposed by Sun et al. [130], we experiment with auxiliary sentences extracted from the 3-class English sentiment dataset with the Sinhala sentiment dataset. By intuition, we hypothesize that such an auxiliary phrase would be able to give an external alignment signal to the model. Compared to Sun et al. [130]'s method we use longer phrases. Since 3-class (*Positive, Neutral, Negative*) sentiment classification is a more generally followed task, we adapt 3-class sentiment classification from here onward for our Sinhala dataset. Thus, we remove the *Conflict* class, which is also an ambiguous class label.

5.3.1.1 Utilizing the lexicons

As mentioned in section 5.3.1 we draw initial ideas from Sun et al. [130]'s method but we devise our method as a two-stage fine-tuning technique. This is due to that CL (Chapter 2) has shown promising results for improving downstream task results.

5.3.2 Steps of the proposed technique

Based on the above initial findings, the steps of our technique are planned as below. Following the auxiliary sentences approach similar to that of Sun et al. [130], we use generated additional phrases which we coin a term *Auxiliary phrases (AP)* . We use a lexicon to generate the APs and utilize them in an intermediate fine-tuning stage as described in the following section. In a summary, the proposed method comprises of two stages,

- We prepend a set of APs from a high-resource lexicon to each training data sample in the target low-resource language dataset. This synthetically augmented dataset creates a binary classification task. In other words, AP and data samples having the same sentiment are labeled as a positive instance (1), or else, a negative instance (0). This binary classification task is considered an intermediate fine-tuning task. We fine-tune a multilingual model for this intermediate task.
- After the fine-tuning with the intermediate task, we further fine-tune the model for the original dataset (without APs) as the final fine-tuning step. We then infer predictions on our test dataset.

5.3.2.1 Intermediate Fine-tuning with Augmented LRL Data

We use a data augmentation method to synthesize the required data for the intermediate fine-tuning task in this step. First, the words for the synthesized APs are selected considering their valence scores provided from the high-resource language lexicon (Figure 5.3). Then, these created APs are prepended to the original data samples of the

target language dataset. The selected MMLM is fine-tuned as an intermediate fine-tuning step with this augmented dataset. Finally, the augmented dataset is discarded and fine-tuning continues only with the low-resource language dataset (see Figure 5.4). The initial hypothesis to follow this step is that we think the auxiliary phrases would be able to perform a certain degree of cross-lingual alignment of the sentiment words, using this intermediate fine-tuning task.

We select appropriate sentiment words (to create the APs) from the high-resource language lexicon as described above. They are then transformed into phrases by considering all the permutations of the selected words. To select the best APs, we use a separate MMLM fine-tuned on a 3-class (*positive, negative, neutral)* sentiment classification dataset of the same high-resource language[1]. As shown in Figure 5.3, we select the best AP(s) by feeding them to the second fine-tuned language model and filtering the phrases that give the highest positive output logit value for the intended sentiment class. Each AP has a specific sentiment based on the words they contain. Our APs resemble a structure similar to “Universal Adversarial Triggers" [131]; however, we use sentiment words from a lexicon to create the APs whereas Wallace et al. [131] create trigger phrases with a refined subset of the model vocabulary (with no reference to sentiment words from an external lexicon).

An augmented dataset is created by prepending APs to the sentences in the target language dataset [2]. When the AP and the target language data sample have a similar sentiment, the augmented sample is labeled as 1, and 0 otherwise. An example of creating and selecting the APs using Sinhala is shown in Figure 5.4. There, terms in the AP contain neutral sentiments (i.e.-valence scores in the (0.4,0.6) interval), which means the AP bears a neutral sentiment. The Sinhala phrases are translated as;*"There's more here than we know"* and *"This work should be given maximum punishment"* which are labeled as neutral and negative (respectively) in the original dataset.

5.4 Implementation details

5.4.1 Datasets and Lexicons

For Sinhala sentiment classification, we use a 3-class version (*Positive, Negative, Neutral)* of the dataset published by Senevirathne et al. [120]. As mentioned previously, we remove the *Conflict* class. For the English dataset, we select a couple of publicly available 3-class datasets (Potts et al. [132] and US Twitter sentiment analysis[3]). For reporting results we have US Twitter sentiment analysis dataset. We also use a German

[1]We create the required fine-tuned model with our English dataset but it could be a different model fine-tuned on the same 3 classes. This fine-tuning is a one-time task.

[2]From experiments, we found that prepending provides slightly better results than appending APs to sentences

[3]https://www.kaggle.com/crowdflower/twitter-airline-sentiment

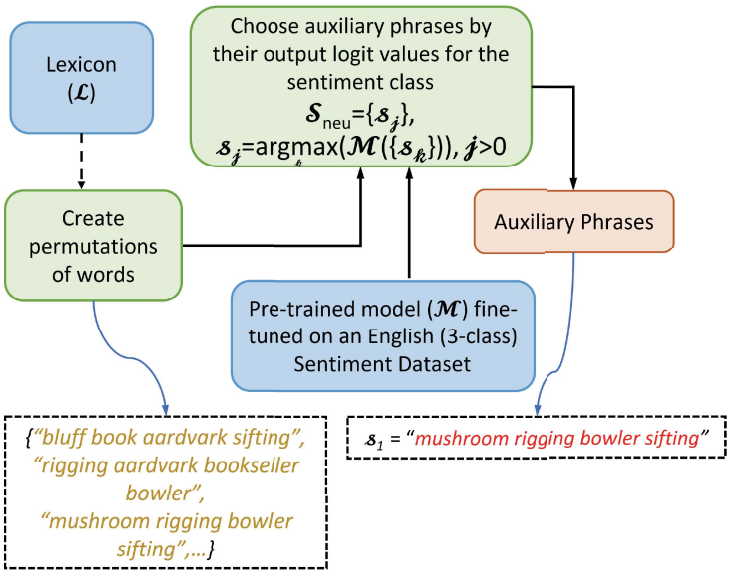

Fig. 5.3: Selecting APs.

ternary dataset (from GermEval-2017 task) which is also considered as a high-resource language[4]. Additionally, we use Vietnamese (from VLSP-2016 task[5]), Tamil [133] and Bengali [134] language datasets. They can be considered as mid resource languages (category 4 and 3 respectively according to Joshi et al. [8]'s categorization of languages).

For the lexicons we mainly utilize VAD sentiment lexicon [91] that contains 20 000 sentiment words with their valence, arousal and dominance scores (continuous values) and VADER [92] which contains 7520 sentiment words/emoticons. For further experiments we use a publicly available Sinhala sentiment lexicon[6].

To identify *positive, neutral, negative* words in the lexicon we first manually define valence score intervals. We verify this manual selection by providing a set of APs in English to an English fine-tuned model and observing that the model predicts the expected sentiment classes. As an example for creating APs, we choose the 3 most positive words from lexicon (e.g.- VADER); *magnificently, ilu, aml* and create permutations from them. The permutations are then fed into a fine-tuned model and the best is selected by the highest logit value output for *positive* sentiment class prediction. In this example, we expect negative, neutral and positive predictions at indexes 1,2 and 3 respectively from the model output array. Hence, here we choose the 4th permutation in the list.

1. *aml magnificently ilu*: [-2.0061314, -1.4377168, 3.1962798]

[4]https://sites.google.com/view/germeval2017-absa
[5]https://vlsp.org.vn/resources-vlsp2016
[6]https://github.com/binodmx/helasentilex

2. *aml ilu magnificently*: [-1.8522748, -1.5239806, 3.1405883]

3. *magnificently aml ilu*: [-2.0096264, -1.4296048, 3.1805775]

4. *magnificently ilu aml*: [-1.9999465, -1.4706941, **3.1985717**]

5. *ilu aml magnificently*: [-1.6413125, -1.6105448, 3.0310764]

6. *ilu magnificently aml*: [-1.9787084, -1.4326444, 3.1722727]

Below we present some example APs produced by the two lexicons we use in our experiments.

VAD lexicon

- Positive - *very positive magnificent love happy, joyful greatness happiest happier*
- Neutral - *aardvark bluff bookseller token, mushroom rigging bowler sifting*
- Negative - *shit suffering died toxic, decayed pain murderer chaos*

VADER

- Positive - *magnificently ilu aml, euphoria ecstacy hearts sweetheart*
- Neutral - *borer skeptics %)*
- Negative - *slavery raping rapist, murder rape kill terrorist*

5.4.2 Experimentation setup and Baseline

We first obtain results for each language dataset under a vanilla fine-tuning using XLM-R-base model. Then, zero-shot results are also obtained for the dataset by using an XLM-R-base model which was fine-tuned on our English (Tweets) dataset. Next, for our newly proposed technique, we form an intermediate binary classification dataset for each language using the lexicons. There we randomly select 50% of the original dataset sentences/documents to be prepended by APs which creates positive instances and the other half to be negative instances.

For all the experiments, we use a single GPU resource which is similar to the one described in Chapter 4. The code implementations are carried out using Tensorflow and Pytorch. For reporting the results we use macro-F1 score averaged across 3 randomly initialized runs. We present the hyperparameters used in Table 5.6. AdamW [122] was used for all fine-tuning tasks.

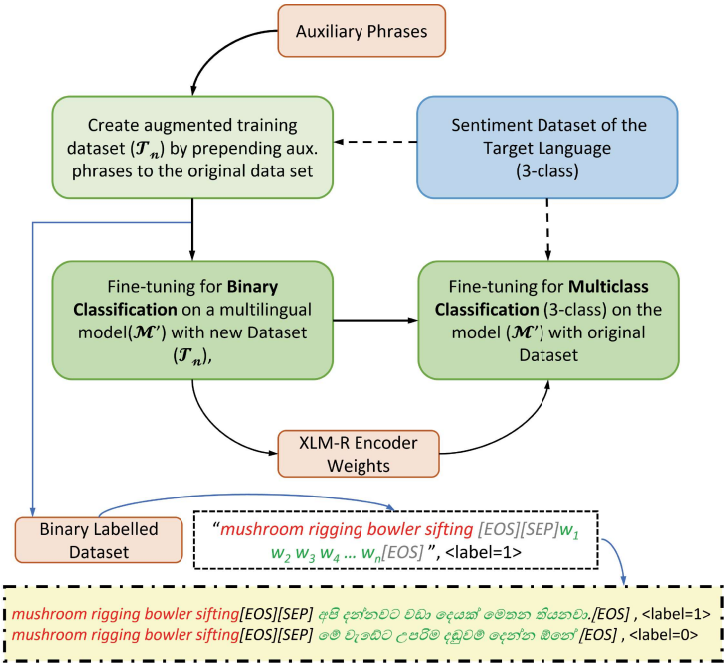

Fig. 5.4: Two stage fine-tuning method proposed.

Dataset	Train/Test	Epochs	Learning rate	Batch size
English	13176/1464	3, 1	5e-6	16
German	20941/2566	2, 3	"	16
Sinhala	11833/1314	4, 3	"	"
Tamil	15694/1743	3, 2	"	"
Bengali	14853/3000	5, 4	"	"
Vietnamese	4123/1050	1, 4	"	"

TABLE 5.3: Parameters used for each dataset

5.5 Results

Table 5.4 reports the baseline, zero-shot and the results obtained from the proposed method. There, zero-shot refers to the scenario that we vanilla fine-tune XLM-R model with English dataset and predict class labels for other datasets.

5.5.1 Ablation studies

In order to to analyze the output with the change different variable parameters in our proposed technique, we carry out several experiments as an ablation study. For all the experiments, we use the Sinhala sentiment dataset. We identify below as the variable parameters in our ablation study.

- Number of different APs used

Dataset	Baseline	Zero-shot	Our Method
English	80.17	-	**81.32** (+1.15)
German	65.56	48.29	**66.66** (+1.10)
Sinhala	69.61	62.23	**71.19** (+1.58)
Tamil	63.87	43.46	**64.67** (+0.80)
Bengali	42.73	40.20	**43.26** (+0.33)
Vietnamese	71.33	57.61	**71.69** (+0.36)

TABLE 5.4: macro-F1 scores of experiments comparing our method against each language's baseline on XLM-R (base)

- Number of words in a single AP
- Language of the APs
- Lexicon used

We then observe the outputs by changing the above parameters. For the number of words in an AP experiment (2nd variable in the list) a single AP was used. For experiments with the first two variables in the above list, we create APs in the English language. Figure 5.5 shows the output results (macro-F1) for the two experiments.

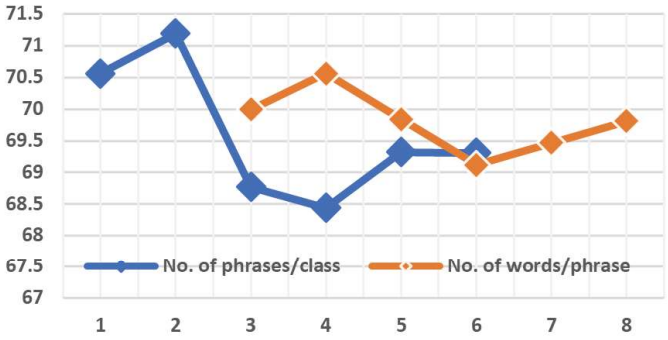

Fig. 5.5: macro-F1 scores for the results by changing the no. of APs and no. of word

In order to change the language of AP, we select from the set of English, Hindi, Tamil, Bengali, and Sinhala. There, we keep to a number of APs as 2 and 4 words maximum in an AP, following the results from the previous two ablation experiments. Additionally, we create random APs in English using randomly picked words from the English lexicon (without following our procedure of identifying the best APs). For changing the lexicon, we choose from VAD sentiment lexicon and VADER. Table 5.5 shows the results for the rest of the varying parameters.

Finally, in order to get a glimpse of the effect of our technique on the embedding space of the model, we try to visualize embedding vectors of some sentiment words

AP attribute	F1
Baseline - vanilla fine-tuning	69.61
1. APs in different languages	
English	**70.56**
Sinhala	69.66
Tamil	70.28
Bengali	69.16
Hindi	69.73
2. Randomly selected APs	67.66
3. APs created with different lexicons	
VAD Sentiment Lexicon	**70.56**
VADER	69.99

TABLE 5.5: Results for experiments with varying attributes of the APs for Sinhala sentiment dataset.

in Sinhala and English. We hypothesize that the sentiment word embeddings for Sinhala words should be weakly positioned (i.e.- not aligned properly according to their meanings) in the embedding space. We observe how the word embeddings created by the XLM-R-base model for several sentiment words change when our method is used. [CLS] token's representation was used as the word vector for each word. We select 16 *positive* and *negative* words in English and Sinhala languages, which are also present in our training data and the VAD lexicon. We perform a dimensionality reduction using Truncated Singular Value Decomposition (Truncated SVD) followed by t-SNE [135] to obtain 2 dimensional (2D) representations of original XLM-R embeddings for the words. For dimensionality reduction, we set a fixed random state (We use Scikit-Learn's implementation[7]) and try with different perplexity values for t-SNE in [1, 50] interval and choose the visualization producing the lowest Kullback-Leibler (KL) divergence after 1000 iterations (perplexity value=10). The obtained visualization is shown in Figure 5.6.

5.6 Adapter based experiments

We also trained a couple of adapters (language and task) based on Pfeiffer et al. [2] adapter architecture, which we integrate into our proposed two-stage method. We include each newly trained adapter in the XLM-R model and fine-tune further with our proposed method to observe any improvement in results. We newly train an English language adapter based on the 3-class English sentiment dataset we use in our experiments as well as task adapters for sentiment classification using the same English dataset and Sinhala 3-class sentiment dataset. The main idea was to benefit from the additional parameters introduced by adapters to the model. we train a language adapter

[7]https://scikit-learn.org

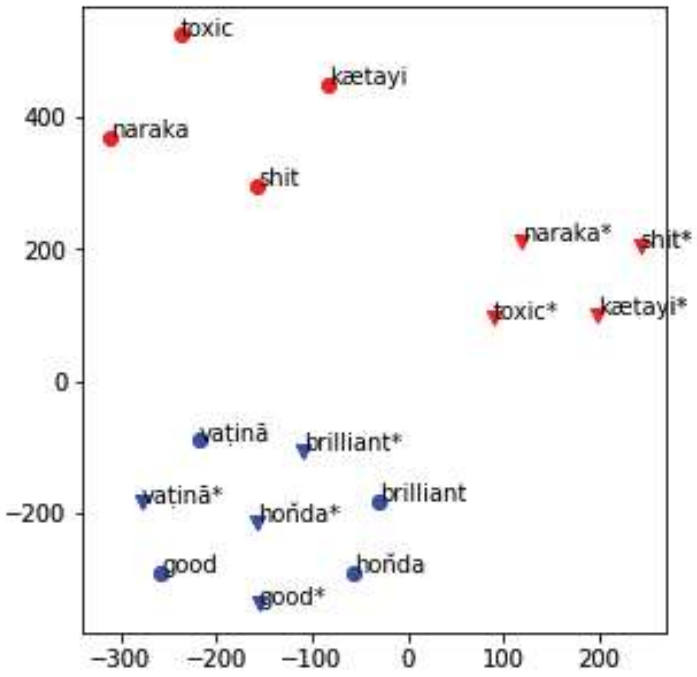

Fig. 5.6: Visualization of XLM-R-base word embeddings without applying our technique (circle markers) and after applying our technique (triangle markers). The blue markers represent positive sentiment words and red markers represent negative sentiment words.

in English primarily considering it would help the model with high-resource language knowledge. However, we do not observe improved results. This could be due to that, even though we introduce additional parameters to the model by the means of adapters, the model does not get to train (or fine-tune) with new data, as we use the same datasets for adapter creation. Since the results are lower in the initial experiments, we do not continue with all the possible experiments with adapters for the datasets.

Dataset	w/ En Lang. adapter	w/ En task adapter	w/ Si lang. adapter
German	42.92	-	-
Sinhala	69.54	70.19	69.52
Tamil	44.64	-	-
Vietnamese	55.00	-	-

TABLE 5.6: Macro-F1 scores from adapter based experiments with the proposed method

5.7 Discussion

The results in section 5.5 shows that the proposed method do not consistently perform across all the languages. Yet, it shows comparatively high results for English, German and Sinhala. We hypothesize the output results vary based on two factors, which are, *the quality of the target language dataset or its baseline performance*, and the *similarity of the target language to the language of APs (English)*. Using a tool such as *lang2vec*[8] [136] language similarities can be numerically quantified. With the aforementioned tool, based on their genetic (phylogenetic) similarity to English the languages can be listed as below. This can be also considered according to the language family, where

1. German
2. Bengali
3. Hindi
4. Sinhala
5. Tamil
6. Vietnamese

In the results obtained, it can be observed that the high results for German can directly be due to its closeness to English (in both lists). Although Sinhala is more distant from English than languages like Bengali or German are, it shows good performance with our method. This could be due to that the better baseline performance of the Sinhala dataset that we use. In other words, we think the low results occur for datasets such as Bengali as their baseline performance was weaker than that of Sinhala, even though they are more similar to English .

In our set of ablation experiments, we observe that not all the auxiliary phrase constructions contribute to result improvement. This could be mainly due to that the model tends to overfit when the external information carried in the auxiliary phrases is not at the optimum level. This is reflected by the experiments of varying the number of APs used and the number of words per AP experiments. In our experiments, we find maximum two APs with 4 sentiment words per each AP is provides the highest gains observed. Unsurprisingly, using English as the language of APs, yield the best results. This shows that using high-resource languages such as English can benefit model fine-tuning for low-resource language datasets. Using random words to create APs degrades the results as we can expect randomly picked words to hinder the

[8] https://github.com/antonisa/lang2vec/

strength of the sentiment value contained in an AP. In other words, an AP created with randomly picked words would not produce a clear external signal to the model whereas our proposed method focuses on creating APs that have only relevant sentiment words belonging to a particular sentiment class. Among the two lexicons, the VAD sentiment lexicon produces better results. This could be due to that the VAD sentiment lexicon offers more words with fine-grained sentiment values. Moreover, using From t-SNE visualization we observe that our proposed method performs an alignment of sentiment words to a certain extent. We assume that our proposed method can perform a change of word vectors' position in the latent space of the XLM-R model. Further probing and experimentation could help to identify at which model layers this process takes place. It would further help to optimize the proposed method and also to understand how the word vectors are positioned in multilingual vector space. It also consolidates the idea of how the model identifies words/text bearing similar meanings based on how they are positioned in the vector space.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Transformer-based pre-trained language models have become a standard tool for modern NLP tasks. However, the under-representation of low-resource languages like Sinhala in these models is a challenge to overcome considering many aspects. Transformer based multilingual language models offer a workaround to this problem. They include multiple languages in a single model and low-resource languages can also benefit from transfer learning. Yet, it is still debatable whether monolingual or multilingual models are the best for any given language-task pair [53, 59, 62]. In this research, we first compare the performance of a few such Transformer based monolingual and multilingual language models for a set of Sinhala text-classification tasks including sentiment analysis, news source classification, news topic classification, and writing style classification. We also pre-train two new Sinhala monolingual models (“SinBERT") based on RoBERTa, and a large Sinhala monolingual corpus. We identify that the multilingual XLM-R-large model consistently stays ahead of the other models for all the classification tasks. This is due to the large amounts of data used to pre-train the multilingual model, the high number of parameters in the model (high model capacity), and the transfer learning that takes place within the model. This result is in line with many findings such as However, our newly pre-trained monolingual models show better performance surpassing the XLM-R-base model for low dataset sizes thus proving smaller monolingual models are useful compared to the large multilingual models.

In the following part of our research, we focus on devising a novel method to improve downstream task performance for a sentiment analysis task in Sinhala. There, we primarily focus on the multilingual XLM-R-base model. First, we experiment with some techniques and a few of their variants that already exist in the literature such as data augmentation, adapters, etc. However, these experiments do not yield consistent improvements (mainly) on our Sinhala dataset and some of these methods are complex to implement practically as well [85, 100]. We propose a technique that uses an external knowledge base (lexicon) of a high-resource language like English and a two-stage fine-tuning method. Our method performs a cross-lingual alignment between the high-resource language and the target low-resource language. We further experiment with a few other languages which are high, mid, or low-resource languages, and observe varying gains for each of them. We think that these varying gains are based on the language similarity and the quality of the dataset.

Continuing from the first part of our findings, we expect that our newly pre-trained models can be further experimented on other NLU tasks such as NER such that new

benchmarks can be produced for NLU datasets in Sinhala as future work. For the second part of our research we can further test the proposed method on other language datasets and other models as we primarily used XLM-R model in our work.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf viii, 1, 4, 6, 8

[2] J. Pfeiffer, I. Vulić, I. Gurevych, and S. Ruder, "MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 7654–7673. [Online]. Available: https://aclanthology.org/2020.emnlp-main.617 viii, 13, 20, 37, 38, 47

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423 1, 2, 7, 10

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692 1, 9

[5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683 1

[6] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. [Online]. Available: https://aclanthology.org/W18-5446 1

[7] A. Conneau, R. Rinott, G. Lample, A. Williams, S. Bowman, H. Schwenk, and V. Stoyanov, "XNLI: Evaluating cross-lingual sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2475–2485. [Online]. Available: https://aclanthology.org/D18-1269 1, 35

[8] P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury, "The state and fate of linguistic diversity and inclusion in the nlp world," *arXiv preprint arXiv:2004.09095*, 2020. 1, 43

[9] Y. K. Lal, R. Singh, H. Trivedi, Q. Cao, A. Balasubramanian, and N. Balasubramanian, "IrEne-viz: Visualizing energy consumption of transformer models," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 251–258. [Online]. Available: https://aclanthology.org/2021.emnlp-demo.29 1

[10] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4996–5001. [Online]. Available: https://aclanthology.org/P19-1493 1, 2, 12

[11] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747 2, 9, 17, 18

[12] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," *CoRR*, vol. abs/2010.11934, 2020. [Online]. Available: https://arxiv.org/abs/2010.11934 2

[13] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov.

2019, pp. 6382–6388. [Online]. Available: https://aclanthology.org/D19-1670 2, 20, 37

[14] J. Ács, Á. Kádár, and A. Kornai, "Subword pooling makes a difference," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2284–2295. [Online]. Available: https://aclanthology.org/2021.eacl-main.194 2

[15] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4487–4496. [Online]. Available: https://aclanthology.org/P19-1441 2, 14, 20, 36

[16] F. M. P. del Arco, S. Halat, S. Padó, and R. Klinger, "Multi-task learning with sentiment, emotion, and target detection to recognize hate speech and offensive language," *CoRR*, vol. abs/2109.10255, 2021. [Online]. Available: https://arxiv.org/abs/2109.10255 2

[17] V. Dhananjaya, P. Demotte, S. Ranathunga, and S. Jayasena, "Bertifying sinhala – a comprehensive analysis of pre-trained language models for sinhala text classification," 2022. [Online]. Available: https://arxiv.org/abs/2208.07864 3

[18] M. Artetxe and H. Schwenk, "Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 597–610, 2019. [Online]. Available: https://aclanthology.org/Q19-1038 4, 11, 35

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: https://aclanthology.org/N18-1202 4

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 17351780, nov 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735 4

[21] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555 4

[22] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *CoRR*, vol. abs/1610.10099, 2016. [Online]. Available: http://arxiv.org/abs/1610.10099 5

[23] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *CoRR*, vol. abs/1705.03122, 2017. [Online]. Available: http://arxiv.org/abs/1705.03122 5

[24] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165 5

[25] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT sentence embedding," *CoRR*, vol. abs/2007.01852, 2020. [Online]. Available: https://arxiv.org/abs/2007.01852 8

[26] L. Pan, C.-W. Hang, H. Qi, A. Shah, S. Potdar, and M. Yu, "Multilingual BERT post-pretraining alignment," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 210–219. [Online]. Available: https://aclanthology.org/2021.naacl-main.20 8

[27] Z. Chi, L. Dong, F. Wei, N. Yang, S. Singhal, W. Wang, X. Song, X.-L. Mao, H. Huang, and M. Zhou, "InfoXLM: An information-theoretic framework for cross-lingual language model pre-training," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 3576–3588. [Online]. Available: https://aclanthology.org/2021.naacl-main.280 8

[28] S. Cao, N. Kitaev, and D. Klein, "Multilingual alignment of contextual word representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=r1xCMyBtPS 8

[29] H. Huang, Y. Liang, N. Duan, M. Gong, L. Shou, D. Jiang, and M. Zhou, "Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2485–2494. [Online]. Available: https://aclanthology.org/D19-1252 8

[30] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, sep 2020. [Online]. Available: https://doi.org/10.10072Fs11431-020-1647-3 8, 12, 14

[31] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *CoRR*, vol. abs/1901.02860, 2019. [Online]. Available: http://arxiv.org/abs/1901.02860 9

[32] R. Sennrich, B. Haddow, and A. Birch, "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)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://aclanthology.org/P16-1162 9

[33] A. CONNEAU and G. Lample, "Cross-lingual language model pretraining," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf 9, 10

[34] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 66–75. [Online]. Available: https://aclanthology.org/P18-1007 9

[35] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012 9

[36] S. Wu and M. Dredze, "Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 833–844. [Online]. Available: https://aclanthology.org/D19-1077 12, 20

[37] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 1820, 2019, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2019, p. 194206. [Online]. Available: https://doi.org/10.1007/978-3-030-32381-3_16 12, 13, 20

[38] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *CoRR*, vol. abs/1705.04228, 2017. [Online]. Available: http://arxiv.org/abs/1705.04228 12

[39] Z. Wang, W. Bi, Y. Wang, and X. Liu, "Better fine-tuning via instance weighting for text classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 7241–7248, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4709 12

[40] I. Li, P. Sen, H. Zhu, Y. Li, and D. Radev, "Improving cross-lingual text classification with zero-shot instance-weighting," in *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1–7. [Online]. Available: https://aclanthology.org/2021.repl4nlp-1.1 12

[41] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2790–2799. [Online]. Available: https://proceedings.mlr.press/v97/houlsby19a.html 13, 37

[42] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," ser. Psychology of Learning and Motivation, G. H. Bower, Ed. Academic Press, 1989, vol. 24, pp. 109–165. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079742108605368 13

[43] B. Ermis, G. Zappella, M. Wistuba, and C. Archambeau, "Memory efficient

continual learning for neural text classification," 2022. [Online]. Available: https://arxiv.org/abs/2203.04640 13

[44] Z. Ke, H. Xu, and B. Liu, "Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 4746–4755. [Online]. Available: https://aclanthology.org/2021.naacl-main.378 13

[45] Z. Ke, B. Liu, H. Wang, and L. Shu, "Continual learning with knowledge transfer for sentiment classification," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 1418, 2020, Proceedings, Part III*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 683698. [Online]. Available: https://doi.org/10.1007/978-3-030-67664-3_41 13

[46] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 328–339. [Online]. Available: https://aclanthology.org/P18-1031 13

[47] D. Hershcovich, S. Frank, H. Lent, M. de Lhoneux, M. Abdou, S. Brandl, E. Bugliarello, L. C. Piqueras, I. Chalkidis, R. Cui, C. Fierro, K. Margatina, P. Rust, and A. Søgaard, "Challenges and strategies in cross-cultural nlp," 2022. [Online]. Available: https://arxiv.org/abs/2203.10020 13

[48] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 440–447. [Online]. Available: https://aclanthology.org/P07-1056 13

[49] M.-W. Chang, L. Ratinov, D. Roth, and V. Srikumar, "Importance of semantic representation: Dataless classification," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'08. AAAI Press, 2008, p. 830835. 14

[50] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *CoRR*, vol. abs/1802.07569, 2018. [Online]. Available: http://arxiv.org/abs/1802.07569 14

[51] Z. Ke, H. Xu, and B. Liu, "Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 4746–4755. [Online]. Available: https://aclanthology.org/2021.naacl-main.378 14

[52] Z. Ke, B. Liu, H. Wang, and L. Shu, "Continual learning with knowledge transfer for sentiment classification," *ArXiv*, vol. abs/2112.10021, 2020. 14

[53] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "Flaubert: Unsupervised language model pre-training for french," *CoRR*, vol. abs/1912.05372, 2019. [Online]. Available: http://arxiv.org/abs/1912.05372 16, 17, 18, 51

[54] L. Martin, B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT: a tasty French language model," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7203–7219. [Online]. Available: https://aclanthology.org/2020.acl-main.645 16, 17, 18

[55] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, and V. Basile, "Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets," in *Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15, 2019*, ser. CEUR Workshop Proceedings, R. Bernardi, R. Navigli, and G. Semeraro, Eds., vol. 2481. CEUR-WS.org, 2019. [Online]. Available: http://ceur-ws.org/Vol-2481/paper57.pdf 16

[56] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for chinese bert," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3504–3514, 2021. 16

[57] W. Antoun, F. Baly, and H. M. Hajj, "Arabert: Transformer-based model for arabic language understanding," *CoRR*, vol. abs/2003.00104, 2020. [Online]. Available: https://arxiv.org/abs/2003.00104 16

[58] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo, "Multilingual is not enough: BERT for finnish," *CoRR*, vol. abs/1912.07076, 2019. [Online]. Available: http://arxiv.org/abs/1912.07076 16, 17

[59] D. Q. Nguyen and A. Tuan Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1037–1042. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.92 17, 51

[60] S. Ralethe, "Adaptation of deep bidirectional transformers for Afrikaans language," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 2475–2478. [Online]. Available: https://aclanthology.org/2020.lrec-1.301 16

[61] S. Wu and M. Dredze, "Are all languages created equal in multilingual BERT?" in *Proceedings of the 5th Workshop on Representation Learning for NLP*. Online: Association for Computational Linguistics, Jul. 2020, pp. 120–130. [Online]. Available: https://aclanthology.org/2020.repl4nlp-1.16 17, 18, 19

[62] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. M. Khapra, and P. Kumar, "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4948–4961. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.445 18, 21, 51

[63] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," *CoRR*, vol. abs/1909.11942, 2019. [Online]. Available: http://arxiv.org/abs/1909.11942 18

[64] U. Khalid, M. O. Beg, and M. U. Arshad, "RUBERT: A bilingual roman urdu BERT using cross lingual transfer learning," *CoRR*, vol. abs/2102.11278, 2021. [Online]. Available: https://arxiv.org/abs/2102.11278 18

[65] J. Ács, D. Lévai, and A. Kornai, "Evaluating transferability of bert models on uralic languages," *arXiv preprint arXiv:2109.06327*, 2021. 19

[66] Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for russian language," *arXiv preprint arXiv:1905.07213*, 2019. 19

[67] S. Doddapaneni, G. Ramesh, A. Kunchukuttan, P. Kumar, and M. M. Khapra, "A primer on pretrained multilingual language models," *CoRR*, vol. abs/2107.00676, 2021. [Online]. Available: https://arxiv.org/abs/2107.00676 19

[68] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson, "XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization," *CoRR*, vol. abs/2003.11080, 2020. [Online]. Available: https://arxiv.org/abs/2003.11080 20

[69] A. Lauscher, V. Ravishankar, I. Vulić, and G. Glavaš, "From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4483–4499. [Online]. Available: https://aclanthology.org/2020.emnlp-main.363 19

[70] G. Aguilar, S. Kar, and T. Solorio, "LinCE: A centralized benchmark for linguistic code-switching evaluation," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 1803–1813. [Online]. Available: https://aclanthology.org/2020.lrec-1.223 19

[71] S. Groenwold, S. Honnavalli, L. Ou, A. Parekh, S. Levy, D. Mirza, and W. Y. Wang, "Evaluating the role of language typology in transformer-based multilingual text classification," *CoRR*, vol. abs/2004.13939, 2020. [Online]. Available: https://arxiv.org/abs/2004.13939 19

[72] A. Ebrahimi, M. Mager, A. Oncevay, V. Chaudhary, L. Chiruzzo, A. Fan, J. Ortega, R. Ramos, A. Rios, I. Vladimir, G. A. Giménez-Lugo, E. Mager, G. Neubig, A. Palmer, R. A. C. Solano, N. T. Vu, and K. Kann, "Americasnli: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages," *CoRR*, vol. abs/2104.08726, 2021. [Online]. Available: https://arxiv.org/abs/2104.08726 20

[73] R. Litschko, I. Vulic, S. P. Ponzetto, and G. Glavas, "Evaluating multilingual text encoders for unsupervised cross-lingual retrieval," *CoRR*, vol. abs/2101.08370, 2021. [Online]. Available: https://arxiv.org/abs/2101.08370 20

[74] A. Ebrahimi and K. Kann, "How to adapt your pretrained multilingual model to 1600 languages," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4555–4567. [Online]. Available: https://aclanthology.org/2021.acl-long.351 20

[75] Y. Fujinuma, J. Boyd-Graber, and K. Kann, "Match the script, adapt if multilingual: Analyzing the effect of multilingual pretraining on cross-lingual transferability," 2022. [Online]. Available: https://arxiv.org/abs/2203.10753 20

[76] J. Ma and L. Li, "Data augmentation for chinese text classification using back-translation," *Journal of Physics: Conference Series*, vol. 1651, no. 1, p. 012039, nov 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1651/1/012039 20

[77] M. Bayer, M. Kaufhold, and C. Reuter, "A survey on data augmentation for text classification," *CoRR*, vol. abs/2107.03158, 2021. [Online]. Available: https://arxiv.org/abs/2107.03158 20

[78] Y. Meng, Y. Zhang, J. Huang, C. Xiong, H. Ji, C. Zhang, and J. Han, "Text classification using label names only: A language model self-training approach," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 9006–9017. [Online]. Available: https://aclanthology.org/2020.emnlp-main.724 21

[79] S. H. Muhammad, D. I. Adelani, S. Ruder, I. S. Ahmad, I. Abdulmumin, B. S. Bello, M. Choudhury, C. C. Emezue, S. A. Salahudeen, A. Anuoluwapo, A. Jeorge, and P. Brazdil, "Naijasenti: A nigerian twitter sentiment corpus for multilingual sentiment analysis," *CoRR*, vol. abs/2201.08277, 2022. [Online]. Available: https://arxiv.org/abs/2201.08277 21

[80] D. Aggarwal, V. Gupta, and A. Kunchukuttan, "Indicxnli: Evaluating multilingual inference for indian languages," 2022. [Online]. Available: https://arxiv.org/abs/2204.08776 21

[81] B. Müller, Y. Elazar, B. Sagot, and D. Seddah, "First align, then predict: Understanding the cross-lingual ability of multilingual BERT," *CoRR*, vol. abs/2101.11109, 2021. [Online]. Available: https://arxiv.org/abs/2101.11109 21

[82] J. Singh, B. McCann, R. Socher, and C. Xiong, "BERT is not an interlingua and the bias of tokenization," in *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 47–55. [Online]. Available: https://aclanthology.org/D19-6106 22

[83] S. Wu and M. Dredze, "Do explicit alignments robustly improve multilingual encoders?" in *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, Nov. 2020, pp. 4471–4482. [Online]. Available: https://aclanthology.org/2020.emnlp-main.362 22

[84] Y. Wang, W. Che, J. Guo, Y. Liu, and T. Liu, "Cross-lingual BERT transformation for zero-shot dependency parsing," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5721–5727. [Online]. Available: https://aclanthology.org/D19-1575 22

[85] X. Chen, H. Fan, R. B. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *CoRR*, vol. abs/2003.04297, 2020. [Online]. Available: https://arxiv.org/abs/2003.04297 22, 51

[86] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of Machine Translation Summit X: Papers*, Phuket, Thailand, Sep. 13-15 2005, pp. 79–86. [Online]. Available: https://aclanthology.org/2005.mtsummit-papers.11 22

[87] P. Efimov, L. Boytsov, E. Arslanova, and P. Braslavski, "The impact of cross-lingual adjustment of contextual word representations on zero-shot transfer," 2022. [Online]. Available: https://arxiv.org/abs/2204.06457 22

[88] Z. Liu, G. I. Winata, A. Madotto, and P. Fung, "Preserving cross-linguality of pre-trained models via continual learning," in *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 64–71. [Online]. Available: https://aclanthology.org/2021.repl4nlp-1.8 23

[89] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 64706479. 23

[90] F. Å. Nielsen, "A new ANEW: evaluation of a word list for sentiment analysis in microblogs," *CoRR*, vol. abs/1103.2903, 2011. [Online]. Available: http://arxiv.org/abs/1103.2903 23

[91] S. Mohammad, "Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers). Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 174–184. [Online]. Available: https://aclanthology.org/P18-1017 23, 39, 43

[92] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, pp. 216–225, May 2014. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550 43

[93] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), May 2010. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf 23

[94] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-Based Methods for Sentiment Analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 06 2011. [Online]. Available: https://doi.org/10.1162/COLI_a_00049 23

[95] C. Musto, G. Semeraro, and M. Polignano, "A comparison of lexicon-based approaches for sentiment analysis of microblog posts." in *DART@ AI* IA*. Citeseer, 2014, pp. 59–68. 23

[96] B. Shin, T. Lee, and J. D. Choi, "Lexicon integrated CNN models with attention for sentiment analysis," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 149–158. [Online]. Available: https://aclanthology.org/W17-5220 23

[97] A. Kumar, D. Kawahara, and S. Kurohashi, "Knowledge-enriched two-layered attention network for sentiment analysis," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 253–258. [Online]. Available: https://aclanthology.org/N18-2041 23

[98] K. Margatina, C. Baziotis, and A. Potamianos, "Attention-based conditioning methods for external knowledge integration," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy:

Association for Computational Linguistics, Jul. 2019, pp. 3944–3951. [Online]. Available: https://aclanthology.org/P19-1385

[99] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/12048 23

[100] V. Suresh and D. C. Ong, "Using knowledge-embedded attention to augment pre-trained language models for fine-grained emotion recognition," in *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, pp. 1–8. 23, 39, 51

[101] P. Ke, H. Ji, S. Liu, X. Zhu, and M. Huang, "SentiLARE: Sentiment-aware language representation learning with linguistic knowledge," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6975–6988. [Online]. Available: https://aclanthology.org/2020.emnlp-main.567 23, 40

[102] P. Zhong, D. Wang, and C. Miao, "Knowledge-enriched transformer for emotion detection in textual conversations," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 165–176. [Online]. Available: https://aclanthology.org/D19-1016 23

[103] N. de Silva, "Survey on publicly available sinhala natural language processing tools and research," *CoRR*, vol. abs/1906.02358, 2019. [Online]. Available: http://arxiv.org/abs/1906.02358 23

[104] S. Fernando, S. Ranathunga, S. Jayasena, and G. Dias, "Comprehensive part-of-speech tag set and SVM based POS tagger for Sinhala," in *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 173–182. [Online]. Available: https://aclanthology.org/W16-3718 23

[105] S. Manamini, A. Ahamed, R. Rajapakshe, G. Reemal, S. Jayasena, G. Dias, and S. Ranathunga, "Ananya - a named-entity-recognition (ner) system for sinhala language," in *2016 Moratuwa Engineering Research Conference (MERCon)*, 2016, pp. 30–35. 23

[106] K. Kumarasinghe, G. Dias, and I. Herath, "Sinmorphy: A morphological analyzer for the sinhala language," in *2021 Moratuwa Engineering Research Conference (MERCon)*, 2021, pp. 681–686. 23

[107] D. Upeksha, C. Wijayarathna, M. Siriwardena, L. Lasandun, C. Wimalasuriya, N. de Silva, and G. Dias, "Implementing a corpus for sinhala language," 01 2015. 23

[108] P. Nanayakkara and S. Ranathunga, "Clustering sinhala news articles using corpus-based similarity measures," *2018 Moratuwa Engineering Research Conference (MERCon)*, pp. 437–442, 2018. 24

[109] R. Weerasinghe, D. Herath, and V. Welgama, "Corpus-based Sinhala lexicon," in *Proceedings of the 7th Workshop on Asian Language Resources (ALR7)*. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009, pp. 17–23. [Online]. Available: https://aclanthology.org/W09-3403 24

[110] S. Thillainathan, S. Ranathunga, and S. Jayasena, "Fine-tuning self-supervised multilingual sequence-to-sequence models for extremely low-resource nmt," in *2021 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2021, pp. 432–437. 24

[111] S. Gallege, "Analysis of sinhala using natural language processing techniques," *University of Wisconsin*, 2010. 24

[112] K. Lakmali and P. S. Haddela, "Effectiveness of rule-based classifiers in sinhala text categorization," in *2017 National Information Technology Conference (NITC)*. IEEE, 2017, pp. 153–158. 24

[113] S. Gunasekara and P. S. Haddela, "Context aware stopwords for sinhala text classification," in *2018 National Information Technology Conference (NITC)*. IEEE, 2018, pp. 1–6. 24

[114] N. P. K. Medagoda, "Framework for sentiment classification for morphologically rich languages: A case study for sinhala," Ph.D. dissertation, Auckland University of Technology, 2017. 24

[115] P. Chathuranga, S. Lorensuhewa, and M. Kalyani, "Sinhala sentiment analysis using corpus based sentiment lexicon," in *2019 19th international conference on advances in ICT for emerging regions (ICTer)*, vol. 250. IEEE, 2019, pp. 1–7. 24

[116] P. Jayasuriya, B. Kumarasinghe, S. Ekanayake, R. Munasinghe, S. Thelijjagoda, and I. Weerasinghe, "Sentiment classification of sinhala content in social media," 09 2020. 24

[117] S. Ranathunga and I. U. Liyanage, "Sentiment analysis of sinhala news comments," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 4, pp. 1–23, 2021. 24

[118] P. Demotte, L. Senevirathne, B. Karunanayake, U. Munasinghe, and S. Ranathunga, "Sentiment analysis of sinhala news comments using sentence-state lstm networks," in *2020 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2020, pp. 283–288. 24

[119] Y. Zhang, Q. Liu, and L. Song, "Sentence-state lstm for text representation," *arXiv preprint arXiv:1805.02474*, 2018. 24

[120] L. Senevirathne, P. Demotte, B. Karunanayake, U. Munasinghe, and S. Ranathunga, "Sentiment analysis for sinhala language using deep learning techniques," *arXiv preprint arXiv:2011.07280*, 2020. 24, 27, 31, 35, 42

[121] H. Rathnayake, J. Sumanapala, R. Rukshani, and S. Ranathunga, "Adapter-based fine-tuning of pre-trained multilingual language models for code-mixed and code-switched text classification," *Knowledge and Information Systems*, vol. 64, no. 7, pp. 1937–1966, Jul 2022. [Online]. Available: https://doi.org/10.1007/s10115-022-01698-1 24

[122] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017. 25, 44

[123] N. de Silva, "Sinhala text classification: Observations from the perspective of a resource poor language," 06 2015. 27

[124] D. Sachintha, L. Piyarathna, C. Rajitha, and S. Ranathunga, "Exploiting parallel corpora to improve multilingual embedding based document and sentence alignment," *CoRR*, vol. abs/2106.06766, 2021. [Online]. Available: https://arxiv.org/abs/2106.06766 27

[125] D. Upeksha, C. Wijayarathna, M. Siriwardena, L. Lasandun, C. Wimalasuriya, N. de Silva, and G. Dias, "Implementing a corpus for sinhala language," 01 2015. 28

[126] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019. 30

[127] A. Sarkar, S. Reddy, and R. S. Iyengar, "Zero-shot multilingual sentiment analysis using hierarchical attentive network and bert," in *Proceedings of the 2019*

3rd International Conference on Natural Language Processing and Information Retrieval, 2019, pp. 49–56. 36

[128] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for nlp," *arXiv preprint arXiv:2105.03075*, 2021. 37

[129] M. Wortsman, G. Ilharco, M. Li, J. W. Kim, H. Hajishirzi, A. Farhadi, H. Namkoong, and L. Schmidt, "Robust fine-tuning of zero-shot models," *CoRR*, vol. abs/2109.01903, 2021. [Online]. Available: https://arxiv.org/abs/2109.01903 37

[130] C. Sun, L. Huang, and X. Qiu, "Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 380–385. [Online]. Available: https://aclanthology.org/N19-1035 41

[131] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh, "Universal adversarial triggers for attacking and analyzing NLP," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2153–2162. [Online]. Available: https://aclanthology.org/D19-1221 42

[132] C. Potts, Z. Wu, A. Geiger, and D. Kiela, "Dynasent: A dynamic benchmark for sentiment analysis," *CoRR*, vol. abs/2012.15349, 2020. [Online]. Available: https://arxiv.org/abs/2012.15349 42

[133] A. Hande, S. U. Hegde, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, S. Thavareesan, and B. R. Chakravarthi, "Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced dravidian languages," *CoRR*, vol. abs/2108.03867, 2021. [Online]. Available: https://arxiv.org/abs/2108.03867 43

[134] K. I. Islam, M. S. Islam, and M. R. Amin, "Sentiment analysis in bengali via transfer learning using multi-lingual BERT," *CoRR*, vol. abs/2012.07538, 2020. [Online]. Available: https://arxiv.org/abs/2012.07538 43

[135] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html 47

[136] P. Littell, D. R. Mortensen, K. Lin, K. Kairis, C. Turner, and L. Levin, "URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 8–14. [Online]. Available: https://aclanthology.org/E17-2002 49