Received: 17 May 2023 | Revised: 21 November 2023 | Accepted: 29 November 2023

CAAI Transactions on Intelligence Technology

DOI: 10.1049/cit2.12333

ORIGINAL RESEARCH

Lexicon-based fine-tuning of multilingual language models for low-resource language sentiment analysis

Vinura Dhananjaya[1] | **Surangika Ranathunga**[1,2] | **Sanath Jayasena**[1]

[1]Department of Computer Science and Engineering, University of Moratuwa, Moratuwa, Sri Lanka

[2]School of Mathematical and Computational Sciences, Massey University, Palmerston North, New Zealand

Correspondence

Surangika Ranathunga
Email: s.ranathunga@massey.ac.nz

Funding information

University of Moratuwa, Grant/Award Number: SRC long-term

Abstract

Pre-trained multilingual language models (PMLMs) such as mBERT and XLM-R have shown good cross-lingual transferability. However, they are not specifically trained to capture cross-lingual signals concerning sentiment words. This poses a disadvantage for low-resource languages (LRLs) that are under-represented in these models. To better fine-tune these models for sentiment classification in LRLs, a novel intermediate task fine-tuning (ITFT) technique based on a sentiment lexicon of a high-resource language (HRL) is introduced. The authors experiment with LRLs Sinhala, Tamil and Bengali for a 3-class sentiment classification task and show that this method outperforms vanilla fine-tuning of the PMLM. It also outperforms or is on-par with basic ITFT that relies on an HRL sentiment classification dataset.

KEYWORDS

deep learning, natural languages, natural language processing

1 | INTRODUCTION

Pre-trained multilingual language models (PMLMs) have shown very promising results for text classification, even for low-resource language (LRL) settings [1]. However, there is an imbalance of language representation in these PMLMs—LRLs are severely under-represented in these models (representation is determined by the amount of monolingual data per language used in model pre-training) [2]. Consequently, when fine-tuned with datasets of the same size, results for languages that are well-represented in the models are superior to those of the languages that have a lower representation [3]. The amount of task-specific data used in fine-tuning the PMLMs for downstream tasks is also a deciding factor [3, 4]. However, for LRLs, creating labelled datasets is a challenge. Hence, when using PMLMs for LRL text classification, further improvements should be explored.

Intermediate Task Fine-tuning (ITFT) is a promising technique to improve the performance of PMLMs for downstream tasks, such as sentiment analysis, under resource-poor conditions. In ITFT, the PMLM is first fine-tuned with a dataset from a different language or a different task. Then this model is further fine-tuned with the target task of the considered language [5–7]. We term this basic ITFT.

In contrast to this basic ITFT technique that makes use of a sentiment annotated dataset from a HRL, we propose two intermediate tasks (*TransIT* and *AuxIT*) created from a lexicon belonging to a high-resource language (HRL):

- TransIT—We translate the terms in the HRL lexicon to the LRL using a publicly available Machine Translation system. The (possibly noisy) translations are paired with original lexicon terms to create positive and negative samples, considering their valence scores. Using these samples, we fine-tune the model as an intermediate binary classification task.
- AuxIT—We create a set of synthetic phrases (hereafter referred to as *Auxiliary Phrases* [APs]) using the original HRL lexicon, and prepend them to each training data sample of the target LRL dataset. This augmented dataset helps to create a binary classification task where the AP and data sample having the same sentiment is a positive instance, else, a negative instance. This binary classification task is used as an intermediate task that aligns sentiment words of the target LRL with their counterparts in the HRL.

In both these methods, after the intermediate fine-tuning step is performed, the model is further fine-tuned with the sentiment classification dataset of the target LRL. Both these intermediate fine-tuning methods aim to provide an additional cross-lingual signal to the PMLM, on the relationship between sentiment words belonging to different languages.

Our proposed methods are in line with that of Ke et al. [8]'s objective to include external knowledge, but we use lexicon-based fine-tuning in contrast to their pre-training. For our methods to work, we assume that there exists a sentiment lexicon for an HRL, that has valence scores for each lexicon term. This is not an unreasonable assumption—for English, there are several such lexicons [9, 10].

We experiment with three LRLs, namely Sinhala, Tamil, and Bengali for the task of sentiment classification. We observe that our AuxIT intermediate task outperforms the vanilla fine-tuning baseline. Interestingly, this observation holds for our experiments with an English dataset as well. It also outperforms, or on par with basic ITFT, which makes use of an HRL sentiment classification dataset. Then, we further experiment with sequential ITFT by combining AuxIT with an intermediate task created from a sentiment classification dataset of an HRL (note that this is the type of task used in basic ITFT, as mentioned earlier). In other words, we fine-tune the PMLM in a sequential manner with the new task we propose, as well as the HRL sentiment classification task. Finally, we fine-tune this model with the sentiment classification dataset of the target LRL. This sequential ITFT model further outperforms both vanilla fine-tuning as well as basic ITFT for Sinhala.

We show that the model performance depends on the quality of the HRL lexicon more than the relatedness between the HRL and the target LRL. Interestingly, using a HRL lexicon is better than using a noisy lexicon from the same language, when creating APs using our second method.

2 | RELATED WORK

2.1 | Intermediate task fine-tuning

ITFT has been proposed as a technique that can potentially improve a target task performance on a pre-trained language model (PLM). Phang et al. [11] can be considered as the first to introduce the concept of ITFT for PLMs. They mentioned that ITFT is intended to alleviate catastrophic forgetting of the PLM and improve the robustness of the PLM. However, they further mentioned that determining the proper combination of ITFT tasks and target tasks that work well could be a challenge. As Vu et al. [12] mentioned, factors such as dataset size, the similarity between the source and target tasks, and the domain are important for the effectiveness of ITFT. Pruksachatkun et al. [6] carried out an empirical study on ITFT tasks to understand the mechanisms behind ITFT for cross-task transfer. They mentioned that target tasks that involve reasoning such as question answering, would benefit more from ITFT.

ITFT has been used in both token-level and sequence-level tasks, including tasks that are similar to sentiment classification. As an example of a sequence-level task, in Savini and Caragea [5] a sarcasm detection task was performed using pre-trained BERT-based models. They used multiple intermediate tasks such as emotion detection from general tweets and sentiment classification of movie reviews, and observed that different ITFT tasks can help the target task in different ways. An example of a token-level task is de la Rosa [13] which used ITFT with borrowing word detection as the target task. ITFT has been effective in sequence-sequence tasks such as Neural Machine Translation [14] as well.

2.2 | Use of external knowledge bases

Using external knowledge bases such as lexicons or knowledge graphs has also been proposed as an alternative method to improve the performance of PLMs in downstream NLP tasks. Lauscher et al. [15] proposed a method that can extend BERT to perform better in GLUE benchmark [16] tasks with the help of additionally infused lexical knowledge. Peters et al. [17]'s KnowBERT model, refined with external knowledge using entity-linking modelling and multiple external knowledge sources such as Wikipedia and WordNet, improves performance in tasks such as Word Sense Disambiguation. Similarly, Liu et al. [18] injected PLMs with Wikidata knowledge triplets and showed improved performance for knowledge-intensive downstream tasks.

External linguistic knowledge has been leveraged to improve sentiment classification task results. Teng et al. [19] proposed a simple weighted-sum technique that can leverage lexicons to learn context-aware features for sentiment analysis. Qian et al. [20] showed improved results for sentiment classification with LSTM models, with the help of a lexicon constructed from MPQA [21] and SST dataset[1]. Suresh and Ong [22] proposed a method of using synthesised vector embeddings to provide external knowledge to the model. Particularly, sentiment lexicons have been used as additional knowledge sources to enhance sentiment classification capabilities of deep learning models such as CNNs [23] and RNNs [24–26]. Lexicons have also been used to improve sentiment-aware representations in simple Transformers [27]. Ke et al. [8] proposed a technique that acquires word-level linguistic knowledge into language models such as BERT with the help of a label-aware pre-training task and SentiWordNet [28] to capture sentiment words. However, these works do not directly incorporate ITFT as a method to improve the target task on PLMs.

3 | METHODOLOGY

Our solution is based on ITFT and sentiment lexicons. First, we introduce two intermediate tasks *TransIT* and *AuxIT* created using a sentiment lexicon of an HRL. We fine-tune the PMLM with each of these tasks separately before it is fine-tuned for the

[1] https://nlp.stanford.edu/sentiment/.

sentiment classification task with LRL data. The idea behind introducing such intermediate tasks is to provide an external cross-lingual alignment signal to the model such that the model is facilitated by the understanding of words in the HRL, to improve the understanding of the low-resource ones. Next, we implement sequential ITFT by combining the best of our newly introduced intermediate tasks with an intermediate task created from an HRL sentiment classification dataset.

3.1 | Baselines

We employ three baselines:

- Fine-tune the PMLM with an HRL sentiment classification dataset, and test with the LRL sentiment classification dataset (zero-shot)
- Fine-tune the PMLM with the LRL sentiment classification dataset (i.e. No ITFT)
- Basic ITFT—fine-tune the PMLM first with a sentiment classification dataset of an HRL, and then with the LRL sentiment classification dataset

3.2 | Bilingual sentiment-word phrases as intermediate task data (TransIT)

Our first method is straightforward. We use a sentiment lexicon from an HRL, where each term (i.e. a sentiment word) has a valence score. Valence score can be used as a measure of the sentiment of a word, where the positiveness of a word increases as the valence value nears 1 and negativeness increases when the valence score nears 0 [10, 29].

We create a set of phrases that contain HRL terms and their corresponding translations in the LRL. These terms are selected based on their valence scores (i.e. positive sentiment words corresponding to high valence scores). Then each of these phrases gets labelled as 1, as they carry terms bearing a similar sentiment.

The following example shows how a positive (label 1) is created:

Example: For Tamil; "good *Nalla*[SEP]affection *Pācam* [EOS]"; (label = 1)[2].

We create another set of phrases labelled as 0, by combining original lexical terms with translated terms having a different sentiment:

Example: "good (*Nalla*[SEP]toxic *Naccu*[EOS]"); (label = 0).

Here, the English terms are paired up with a Tamil term with a dissimilar sentiment (the transliterations and translations of the Tamil words in their original script can be found in Figure A2). We use the created phrases in a binary classification task and fine-tune the PMLM.

[2]We used transliterated Tamil words here to avoid script-related issues and improve the paper's readability. But we used the words in their respective scripts in experiments. Translations are shown in Appendix 8.

3.3 | Augmented LRL data as intermediate task data (AuxIT)

Similar to the TransIT method, in this technique as well, auxiliary phrases (or APs) are created considering the valence scores of the HRL lexicon. However, unlike in TransIT, in this method, we create phrases that can be verified to carry the intended sentiment value to the model. We do this by feeding the created phrases to a separately fine-tuned model as a sentiment classification task and ensuring the model classifies the phrases to the expected sentiment class.

We expect that these newly synthesised APs can provide an external alignment signal to the model. In other words, since an AP is guaranteed to have a specific, pre-known sentiment value, we expect that the AP would give an alignment signal related to the particular sentiment class during the intermediate task fine-tuning phase of the model.

To identify *positive*, *neutral* and *negative* words in the lexicon, we first manually define valence score intervals. This is done by manually inspecting the valence score ranges of the lexicon against the words and defining the valence score ranges for *positive*, *neutral* and *negative* words. We verify this manual selection by providing a set of APs in English to a fine-tuned model on the English dataset and observing that the model predicts the expected sentiment classes (an example is shown in the Appendix).

After selecting sentiment words from a lexicon, they are then converted into phrases by considering all the permutations of the selected words. To select the best APs, we use a separate PMLM fine-tuned on a 3-class (*positive*, *negative*, *neutral)* sentiment classification dataset of the same HRL[3]. We define the set of initial APs created using the permutations of sentiment words picked from the lexicon as $\mathbf{S}_k$, where k denotes the sentiment classes ($k \in \{neutral, negative, positive\}$) and $|\mathbf{S}_k| = N, N \in (Z_+)$. The best AP(s) (denoted by the set s) for a particular sentiment class is selected by; $s = \underset{\substack{i \in S_k \\ 1 \le i \le N}}{\text{argmax}} \boldsymbol{M}(i)$, where $\boldsymbol{M}$ denotes the model fine-tuned with English data, and $|s| \geq 1$.

According to the example shown in Figure 1, for the neutral sentiment class, we filter the best AP(s) by feeding them to this fine-tuned PMLM and taking 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" [30]; however, we use sentiment words from a lexicon to create the APs whereas Wallace et al. [30] create trigger phrases with a refined subset of the model vocabulary (with no reference to sentiment words).

Then these APs are prepended to the original data samples of the target language[4]. When the AP and the target language data sample have a similar sentiment, the augmented sample is

[3]We create the required fine-tuned model with our English dataset (Tweets). This could be a different model fine-tuned on the same 3 classes. This fine-tuning is a one-time task.
[4]From initial experiments, we found that prepending provides slightly better results than appending APs to sentences.

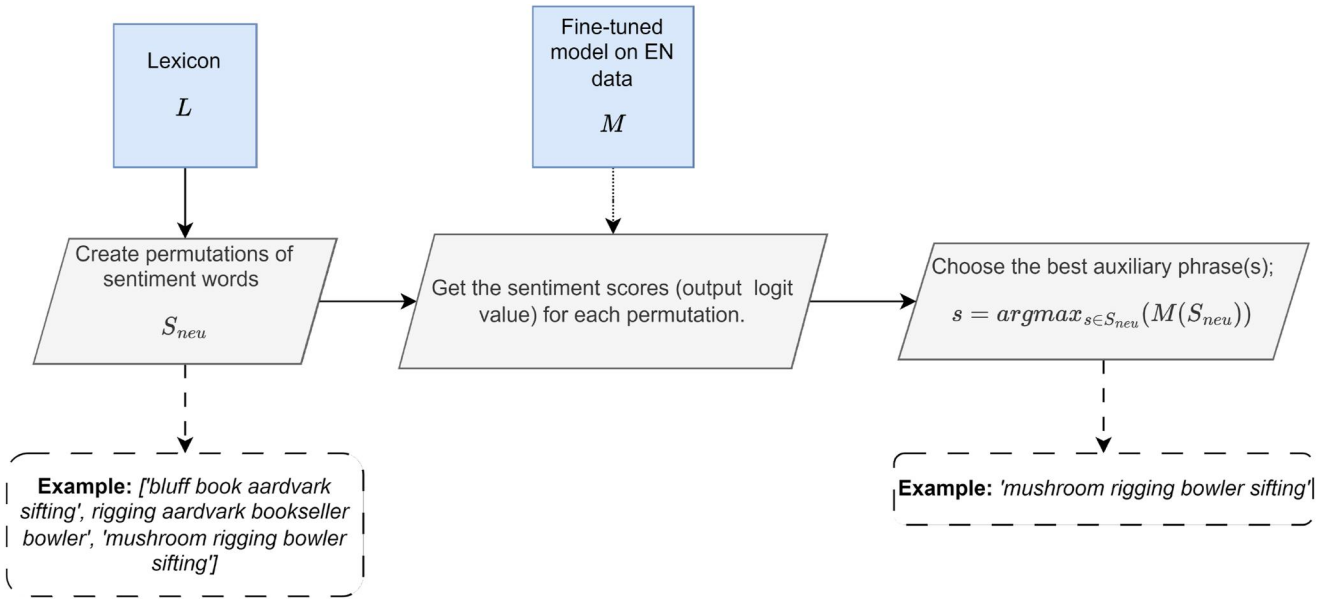

FIGURE 1 Creating APs from the lexicon. A neutral AP is considered as an example. Dotted boxes connected by blue arrows show example instances in respective steps. Dotted arrows represent inputs from lexicons/datasets.

labelled as 1, and 0 otherwise. An example using Sinhala is shown in Figure 2. 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"*.

The PMLM is fine-tuned with this augmented dataset. Finally, this fine-tuned model is again fine-tuned using the LRL dataset as the final training task (see Figure 2).

3.4 | Sequential ITFT

As will be presented in Section 4.3, only our AuxIT intermediate task outperformed the baselines. Therefore we combine this intermediate task with our third baseline (fine-tune the PMLM with data from sentiment classification data of an HRL) in a sequential manner. In other words, we first fine-tune the PMLM with one of these tasks, and then with the other. Finally, the resulting model is further fine-tuned with the dataset from the LRL.

4 | EXPERIMENTS

4.1 | Datasets and lexicons

For English, we use the US Twitter Airline Sentiment dataset[5] (a general domain dataset) and a dataset from the financial domain [31]. Note that English is the HRL used in our experiments. We use a 4-class (*positive*, *negative*, *neutral*, *conflict*) Sinhala sentiment dataset [32], which consists of news comments extracted from news websites, and remove the *conflict* class for our experiments. For Tamil and Bengali, we use datasets released by Hande et al. [33] and Islam et al. [34], respectively (the Tamil dataset consists of code-mixed data samples as well).

We use the VAD sentiment lexicon [10] primarily as our HRL sentiment word lexicon. This lexicon contains valence, dominance, and arousal scores for a set of 20 k English words and their translations for 102 other languages. We also experiment with VADER [9], which consists of 7520 sentiment words (including emojis) and their valence scores.

4.2 | Training setup

We select XLM-R-base as the PMLM. It supports all the languages considered in our experiments and has shown promising results for sentiment analysis [35, 36]. In method 1, we create a binary dataset with 26,000 data samples using all the sentiment words in the VAD lexicon, with 4 terms per data sample. For method 2, we prepend 50% of the original training sentences (to have a balanced number of data points in the two classes) with APs having the same sentiment and the rest with APs having dissimilar sentiments. We average the results across 3 randomly initialised runs and report the macro averages of the F1 scores. Hyperparameters are given in Table A1 in Appendix.

4.3 | Comparative results for different ITFT setups

Table 1 shows the results. Our first method, TransIT yields lower results than the baseline for Sinhala (macro-F1 69.33%) and for Tamil (61.08%). Thus we do not report this result in the table,

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

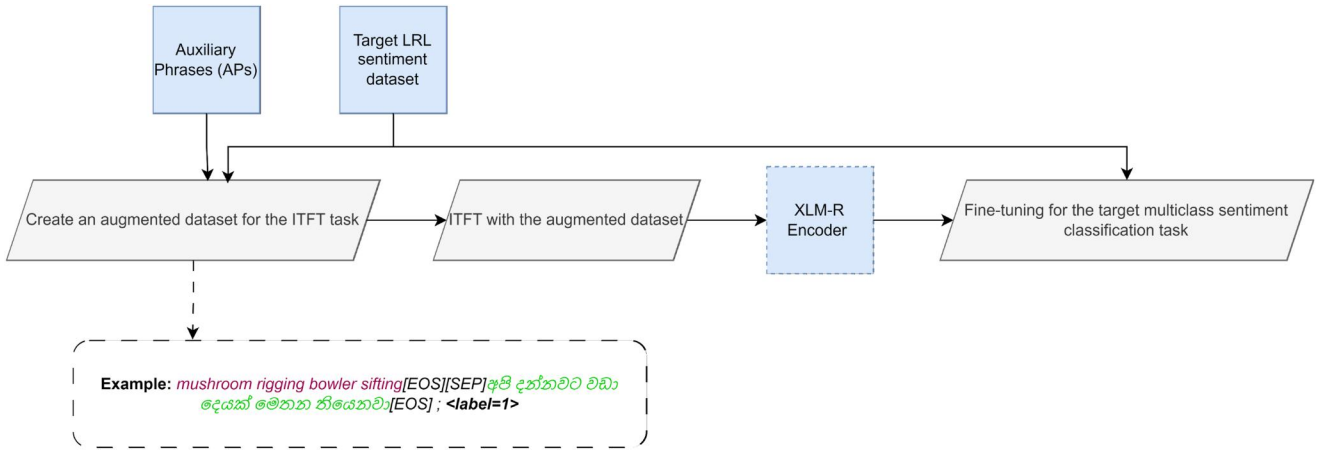

FIGURE 2 Proposed two-stage fine-tuning method using auxiliary phrases. The neutral AP from Figure 1 is considered. An example of using AP with an actual data sample from the Sinhala dataset is shown at the bottom. The phrase in 'red' denotes the created AP which is appended by a sentence of the target LRL dataset (in 'green'). Here, the label = 1 is assigned as both phrases carry the same sentiment value (neutral). The transliterations and translations of the sentences are shown in Figure A1 which are labelled as neutral and negative (respectively) in the original dataset.

TABLE 1 Macro-F1 scores of experiments with different methods. The best performance for each experiment is indicated with bold numbers.

Dataset	Baseline 1	Baseline 2	Baseline 3	AuxIT	Basic ITFT−>AuxIT	AuxIT−>basic ITFT
English (*Tweets*)	-	80.17	-	**81.32**	-	-
English (*Finance*)	-	87.92	-	**88.77**	-	-
Sinhala	62.23	69.61	70.30	71.19	69.10	**71.45**
Tamil	43.46	63.87	64.65	**64.67**	61.97	63.86
Bengali	40.20	42.73	**43.31**	43.26	37.59	39.14

nor do we use TransIT for further experiments. Although the first method is trivial to implement, the phrases created may not always yield the intended sentiment value to the model. This is because, although an AP contains specific sentiment words, it is not guaranteed to convey the intended sentiment through the AP. We verify this by feeding these APs to an XLM-R model fine-tuned on a sentiment classification task and checking their predicted labels. We observe that the model fails to predict the APs to their intended label, even though they contain sentiment words belonging to the respective sentiment class. This could happen because the APs are too short and lack a proper structure to carry useful information [37, 38].

In Table 1, a clear performance gain is visible for our second method (AuxIT) against the second baseline (vanilla Fine-tuning). The highest gains are reported for English and Sinhala datasets. The performance gain of the AuxIT method is better than or on par with basic ITFT (third baseline) as well. Note that for English, which is the HRL considered in our experiments, we use APs from the same language, unlike for the other three languages. Also, basic ITFT does not hold for English, because English is the HRL dataset we used. Interestingly, even for English, the AuxIT method shows noticeable gains, which shows the utility of using lexicons in fine-tuning PMLMs for tasks in HRLs.

In sequential ITFT, when basic ITFT is followed by AuxIT, it did not improve the results. However, the reserve ordering yielded improved results for Sinhala.

5 | ABLATION STUDY

We carry out several ablation experiments on the Sinhala dataset, to determine the effects of the factors given in the following list. We use 1 AP per class for the first three experiments, and compare their results with the baseline obtained from vanilla fine-tuning of XLM-R (see Table 2).

- Effect of the language used to create APs (using the tweets dataset)[6]
- Valence scores of the sentiment words
- Sentiment lexicon
- Number of terms in an AP
- Number of APs

5.1 | Effect of the language used to create APs

To observe the effect of the language (specifically, language relatedness) of the lexicon used to create APs, we experiment with APs created in different languages. We create sentiment lexicons for Hindi, Tamil, and Bengali by translating the VAD

[6]We continue with this dataset as it yielded better gain with our method than the other dataset.

TABLE 2 Results (Macro-F1) for experiments with varying attributes of the APs for Sinhala sentiment dataset. The best performance obtained for each experiment is indicated with bold numbers.

Experiment no.	Changed AP attribute	F1
-	*Baseline 2–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

English lexicon (We used Google Translate). Results are reported in experiment 1 of Table 2. Although Hindi and Bengali belong to the same language family as Sinhala, and Tamil is geographically co-located with Sinhala, the results are low compared to the English lexicon. We believe this is due to the higher representation of English in XLM-R compared to other languages [1]. While translating APs to other languages, translation errors can occur and the noisiness of these translations can be another reason. Some examples of such erroneous translations are presented below. The English words were taken from the VAD lexicon.

- The word *"flop"* (has been given a valence score of 0.081—negative) does not have the correct translation in Sinhala. It only has the transliteration of the word.
- The word *"forge"* (has been given a valence score of 0.52; which is in the neutral region). It has a Sinhala translation which is related to only one of its meanings; "deceptive imitation" which should be a negative sentiment.
- The word phrase *"pissing me off"* (has been given valence score 0.208—negative) is associated with a Sinhala translation with an opposite (positive) sentiment; *"pibidev"* (The English translation of the given Sinhala word *"pibidev"* is *"Arise"*).
- The word *"abbot"* has no translation at all.

Such errors could happen when the lexicon creators try to translate their lexicons into LRLs, relying on machine translation tools at their disposal [39, 40].

5.2 | Effect of the valence scores of lexicon words

To determine the importance of the valence score for creating APs, we created random APs by using randomly picked words from each valence score interval, where we observe a drop in results (experiment 2 in Table 2). This could be due to that randomly picked terms being weak sentiment words (i.e.-valence scores are not strong enough). This observation justifies the AP selection method we introduce in AuxIT.

5.3 | Effect of the sentiment lexicon

Experiment 3 in Table 2 shows the results for the two different lexicons we used (VADER and VAD), where VAD performs well with the possible reason being that it contains a diverse set of words, especially belonging to the neutral class.

5.4 | Effect of the length of APs and number of APs

We also conducted experiments to determine the optimal number of lexicon terms in an AP, and the number of APs used per sentiment class. Figure 3 shows a result drop as the number of words per AP is increased (orange line), possibly due to over-fitting of the model during fine-tuning. We do not experiment beyond 8 words per AP as it takes an excessive amount of time to process and run through all the permutations. We also found that 2 different APs work best for our approach and that more APs could hinder the performance as seen in Figure 3 (blue line).

5.5 | Impact of ITFT on cross-lingual alignment of sentiment words

With the proposed two intermediate fine-tuning methods, we expect to provide an additional cross-lingual signal to the PMLM via the APs. To verify whether our method has been successful in this, we analyse the latent-space representations (word embeddings) of individual sentiment words from Sinhala and English, with, and without our AuxIT intermediate task. We manually pick English and Sinhala positive/negative sentiment words. The visualisation (process details are in Appendix) in Figure 4 shows that Sinhala and English words with similar sentiments have been grouped much closer after the intermediate fine-tuning step. This is particularly true for negative words[7].

6 | CONCLUSION

We proposed two cross-lingual intermediate task fine-tuning methods on PMLMs for sentiment analysis of LRLs, based on a sentiment lexicon of an HRL. Out of these, fine-tuning on augmented data created from the HRL lexicon (AuxIT) yielded noticeable improvements over vanilla fine-tuning. AuxIT outperformed or was on par with basic ITFT as well. We showed

[7] Due to font issues, we show transliterated words in the graph, but use the words in their actual script for experiments. We show their translations and transliterations in Figure A3.

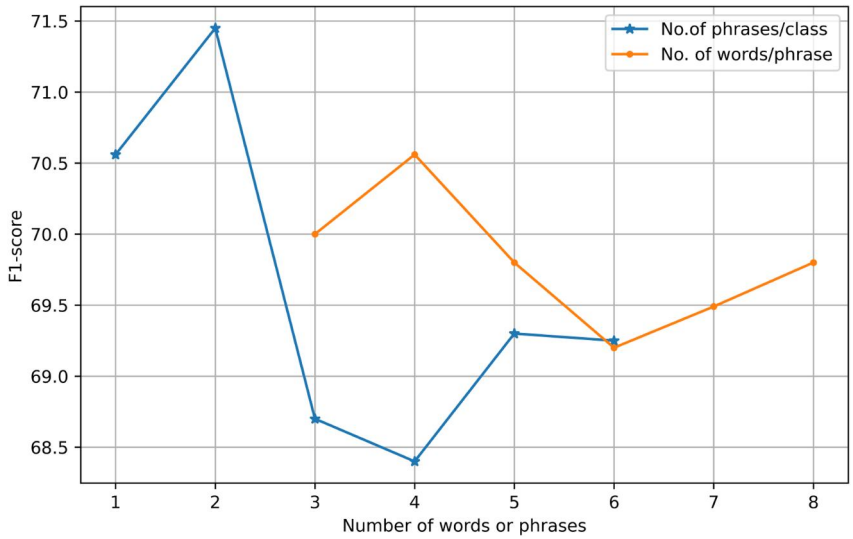

FIGURE 3 macro-F1 score with varying number of APs per class and no. words per AP.

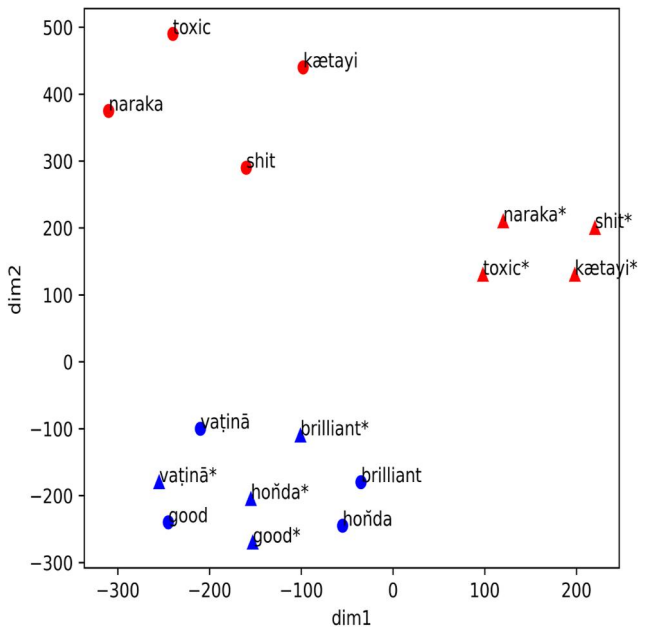

FIGURE 4 Word embeddings visualisation for *positive (blue), negative (red)* words in Sinhala and English. The circle markers show embeddings from a vanilla fine-tuned model (on Sinhala dataset) and the triangle markers show embeddings from our approach.

that this result gain is due to the newly introduced intermediate fine-tuning technique (AuxIT) providing an additional cross-lingual signal to the PMLM to learn the similarity between sentiment words belonging to different languages. We further introduced sequential ITFT, which fine-tunes the PMLM with AuxIT and basic ITFT in a sequential manner.

Our solution was tested only for languages included in XLM-R. In the future, we will consider other models, and languages not included in them. Another future avenue is to refine our method for more fine-grained sentiment classification tasks, whereas, our current experiments considered only a coarse-grained sentiment classification.

ACKNOWLEDGEMENT

Vinura Dhananjaya was funded by a Senate Research Committee grant (SRC/LT/2020/11) of the University of Moratuwa.

Open access publishing facilitated by Massey University, as part of the Wiley - Massey University agreement via the Council of Australian University Librarians.

CONFLICT OF INTEREST STATEMENT

The author declares no conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated during the current study.

ORCID

Surangika Ranathunga https://orcid.org/0000-0003-0701-0204

REFERENCES

1. Hu, J., et al.: Xtreme: a massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In: International Conference on Machine Learning, pp. 4411–4421. PMLR (2020)
2. Ranathunga, S., DeSilva, N.: Some languages are more equal than others: probing deeper into the linguistic disparity in the NLP world. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, pp. 823–848 (2022)
3. Wu, S., Dredze, M.: Are all languages created equal in multilingual BERT? In: Proceedings of the 5th Workshop on Representation Learning for NLP Online, pp. 120–130. Association for Computational Linguistics (2020). https://aclanthology.org/2020.repl4nlp-1.16

4. Doddapaneni, S., et al.: A primer on pretrained Multilingual Language Models. CoRR (2021). abs/2107.00676 https://arxiv.org/abs/2107.00676
5. Savini, E., Caragea, C.: Intermediate-task transfer learning with BERT for sarcasm detection. Mathematics 10(5), 844 (2022). 10.3390/math10050844. https://www.mdpi.com/2227-7390/10/5/844
6. Pruksachatkun, Y., et al.: Intermediate-task transfer learning with pretrained Language Models: when and why does it work? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics Online, pp. 5231–5247. Association for Computational Linguistics (2020). https://aclanthology.org/2020.acl-main.467
7. Chang, T.Y., Lu, C.J.: Rethinking why intermediate-task fine-tuning works. In: Findings of the Association for Computational Linguistics: EMNLP 2021 Punta Cana, Dominican Republic, pp. 706–713. Association for Computational Linguistics (2021). https://aclanthology.org/2021.findings-emnlp.61
8. Ke, P., et al.: SentiLARE: sentiment-aware language representation learning with linguistic knowledge. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) Online, pp. 6975–6988. Association for Computational Linguistics (2020). https://aclanthology.org/2020.emnlp-main.567
9. Hutto, C., Gilbert, E.: VADER: a parsimonious rule-based model for sentiment analysis of social media text. Proceedings of the International AAAI Conference on Web and Social Media 8(1), 216–225 (2014). 10.1609/icwsm.v8i1.14550. https://ojs.aaai.org/index.php/ICWSM/article/view/14550
10. Mohammad, S.: 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), pp. 174–184. Association for Computational Linguistics, Melbourne (2018). https://aclanthology.org/P18-1017
11. Phang, J., Févry, T., Bowman, S.R.: Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-Data Tasks (2018). arXiv–1811.arXiv e-prints
12. Vu, T., et al.: Exploring and predicting transferability across NLP tasks. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) Online, pp. 7882–7926. Association for Computational Linguistics (2020). https://aclanthology.org/2020.emnlp-main.635
13. DelaRosa, J.: ADoBo 2021: the futility of STILTs for the classification of lexical borrowings in Spanish. In: IberLEF@ SEPLN, pp. 947–955 (2021)
14. Nayak, S., et al.: Leveraging Auxiliary Domain Parallel Data in Intermediate Task Fine-tuning for Low-Resource Translation. arXiv preprint arXiv:230601382 2023
15. Lauscher, A., et al.: Informing unsupervised pretraining with external linguistic knowledge. CoRR (2019). abs/1909.02339 http://arxiv.org/abs/1909.02339
16. Wang, A., et al.: 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, pp. 353–355. Association for Computational Linguistics, Brussels (2018). https://aclanthology.org/W18-5446
17. Peters, M.E., et al.: Knowledge enhanced contextual word representations. 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), pp. 43–54. Association for Computational Linguistics, Hong Kong (2019). https://aclanthology.org/D19-1005
18. Liu, L., et al.: Knowledge based Multilingual Language model. CoRR, 10962 (2021). abs/2111 https://arxiv.org/abs/2111.10962
19. Teng, Z., Vo, D.T., Zhang, Y.: Context-sensitive lexicon features for neural sentiment analysis. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1629–1638. Association for Computational Linguistics, Austin (2016). https://aclanthology.org/D16-1169
20. Qian, Q., et al.: Linguistically regularized LSTM for sentiment classification. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1679–1689. Association for Computational Linguistics, Vancouver (2017). https://aclanthology.org/P17-1154
21. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing Vancouver, British Columbia, pp. 347–354. Association for Computational Linguistics, Canada (2005). https://aclanthology.org/H05-1044
22. Suresh, V., Ong, D.C.: 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), pp. 1–8. IEEE (2021)
23. Shin, B., Lee, T., Choi, J.D.: 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, pp. 149–158. Association for Computational Linguistics, Copenhagen (2017). https://aclanthology.org/W17-5220
24. Kumar, A., Kawahara, D., Kurohashi, S.: 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), pp. 253–258. Association for Computational Linguistics, New Orleans (2018). https://aclanthology.org/N18-2041
25. Margatina, K., Baziotis, C., Potamianos, A.: Attention-based conditioning methods for external knowledge integration. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics Florence, Italy, pp. 3944–3951. Association for Computational Linguistics (2019). https://aclanthology.org/P19-1385
26. Ma, Y., Peng, H., Cambria, E.: Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. Proc. AAAI Conf. Artif. Intell. 32(1) (2018). 10.1609/aaai.v32i1.12048. https://ojs.aaai.org/index.php/AAAI/article/view/12048
27. Zhong, P., Wang, D., Miao, C.: 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), pp. 165–176. Association for Computational Linguistics, Hong Kong (2019). https://aclanthology.org/D19-1016
28. Baccianella, S., Esuli, A., Sebastiani, F.: 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) (2010). http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf
29. Mehrabian, A.: Pleasure-arousal-dominance: a general framework for describing and measuring individual differences in temperament. Curr. Psychol. 14(4), 261–292 (1996). https://doi.org/10.1007/bf02686918
30. Wallace, E., et al.: 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), pp. 2153–2162. Association for Computational Linguistics, Hong Kong (2019). https://aclanthology.org/D19-1221
31. Malo, P., et al.: Good debt or bad debt: detecting semantic orientations in economic texts. Journal of the Association for Information Science and Technology 65(4), 782–796 (2014). https://doi.org/10.1002/asi.23062
32. Senevirathne, L., et al.: Sentiment Analysis for Sinhala Language Using Deep Learning Techniques. arXiv preprint arXiv:201107280 2020
33. Hande, A., et al.: Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced dravidian languages. CoRR (2021). abs/2108.03867 https://arxiv.org/abs/2108.03867
34. Islam, K.I., Islam, M.S., Amin, M.R.: Sentiment analysis in Bengali via transfer learning using multi-lingual BERT. In: 2020 23rd International Conference on Computer and Information Technology (ICCIT) IEEE, pp. 1–5 (2020)
35. Rathnayake, H., et al.: Adapter-based fine-tuning of pre-trained multilingual language models for code-mixed and code-switched text

classification. Knowl. Inf. Syst. 64(7), 1937–1966 (2022). https://doi.org/10.1007/s10115-022-01698-1
36. Dhananjaya, V., et al.: BERTifying Sinhala-A comprehensive analysis of pre-trained Language Models for Sinhala text classification. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 7377–7385 (2022)
37. Zhan, J., Dahal, B.: Using deep learning for short text understanding. Journal of Big Data 10(1), 4 (2017). https://doi.org/10.1186/s40537-017-0095-2
38. Hussein, DMEDM: A survey on sentiment analysis challenges. Journal of King Saud University—Engineering Sciences 30(4), 330–338 (2018). 10.1016/j.jksues.2016.04.002. https://www.sciencedirect.com/science/article/pii/S1018363916300071
39. Wan, Y., et al.: Challenges of neural machine translation for short texts. Comput. Ling. 48(2), 321–342 (2022). https://doi.org/10.1162/coli_a_00435
40. Bapna, A., et al.: Building Machine Translation Systems for the Next Thousand Languages.arXiv preprint arXiv:220503983 (2022)
41. VanderMaaten, L., Hinton, G.: Visualizing Data using t-SNE. J. Mach. Learn. Res. 9(86), 2579–2605 (2008). http://jmlr.org/papers/v9/vandermaaten08a.html
42. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)

How to cite this article: Dhananjaya, V., Ranathunga, S., Jayasena, S.: Lexicon-based fine-tuning of multilingual language models for low-resource language sentiment analysis. CAAI Trans. Intell. Technol. 1–10 (2024). https://doi.org/10.1049/cit2.12333

APPENDIX

Visualisation of word embeddings

We look at how XLM-R word embeddings of several sentiment words change when our method is used. We choose *positive* and *negative* words in Sinhala and English, 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 [41] to get 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[8]) and try with different perplexity values for t-SNE in [1, 50] interval and choose the visualisation producing the lowest Kullback-Leibler (KL) divergence after 1000 iterations (perplexity = 10). We used the [CLS] token's representation as the word vector. We select 16 words in both English and Sinhala and the transliterations/translations of selected Sinhala words are shown in Figure A3.

A.1 | Hyperparameters

Table A1 shows hyperparameters and dataset sizes used for each baseline experiment in different languages. Epochs separated by a comma are for the intermediate fine-tuning task and the final 3-class classification task respectively. We use fewer epochs for Tamil than other languages, as we observe higher epochs tend to overfit for the Tamil dataset using our method. We use AdamW [42] for all fine-tuning tasks.

An example for selecting an AP based on the output logit values. We choose the 3 most positive words from the lexicon (e.g. - VADER); *magnificently, ilu, aml* and create

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

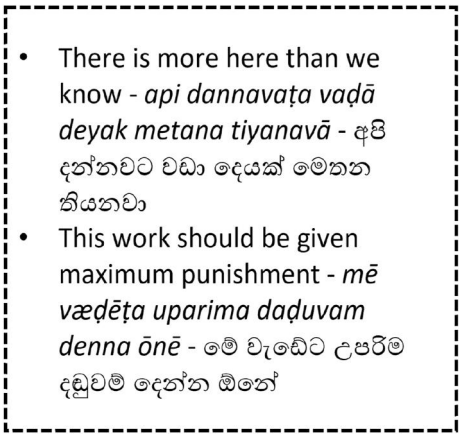

FIGURE A1 Transliterations/translations of the example sentiment sentences shown in Figure 2.

FIGURE A2 Translations of the Tamil words used for the example in Section 3.2 (for Method-1).

FIGURE A3 Transliterations/Translations for Sinhala words used in Figure 4.

TABLE A1 Parameters used for each dataset for getting baseline results 1.

Dataset	Train/Test	Epochs	Learning rate	Batch size
English (*Tweets*)	13176/1464	4, 3	5e-6	16
English (*Finance*)	2037/1315	2, 3	5e-6	8
Sinhala	11833/1314	4, 3	"	"
Tamil	15694/1743	3, 2	"	"
Bengali	14853/3000	5, 4	"	"

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 fourth permutation in the list.

1. *aml magnificently ilu*: [-2.0061314, −1.4377168, 3.1962798]
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, 3.0310,764]
6. *ilu magnificently aml*: [-1.9787084, −1.4326444, 3.1722727]

APs (top two in each sentiment class) created using 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*

APs(top two in each sentiment class) created using VADER lexicon

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

Computational resources

For all our experiments, we used the XLM-R-base model which contains 270 M parameters, as the multilingual model. We utilised a single shared GPU (Nvidia Quadro RTX 6000 24 GB). On average, it consumes ~0.4 h for one randomly initialised run in each experiment.