Meta-Learning for One-Shot Low-Resource Sentiment Analysis

Andy Lee, Lokesh Sreenath, Tong Wang

Abstract

Low-resource languages present a strong challenge for text classification tasks due to the small amount of available data. Some languages do not have built in support in many applications. Models which rely on large corpuses are ineffective in performing these tasks. Twitter is widely used globally and provides a rich source of data for many languages. Often tweets will contain multilingual text as well, providing an additional challenge. We propose tackling the problem of sentiment analysis on a variety of low-resource languages using meta-learning. Our overall architecture involves a pipeline of a multilingual embedding model (LASER), a data augmentation model (SuperGen), and model agnostic machine learning (MAML). We aim to predict both the sentiment as well as the language from a variety of tweet datasets with provided ground truth annotations. These languages span high resource languages, dialects of high resource languages, and low resource languages. Our findings indicate that MAML, a common meta-learning method, may not be effective for 1-shot sentiment analysis of low resource languages.

1 Introduction

Low resource languages provide a unique challenge for text classification. These languages have small quantities of existing data, may have skewed source of data, and may not be easily manually annotated. Many approaches have been suggested for text classification tasks on these languages. These approaches include utilizing pre-trained language models, data augmentation, and classical NLP methods.

Here we are specifically interested in the one-shot case for multilingual sentiment analysis. We develop a framework centered around Model Agnostic Meta-Learning (MAML) [9] which provides a training framework for N-Way K-Shot learning. We try two different embedding algorithms, FastText and LASER, to convert a corpus of annotated Tweets from a variety of languages into text embeddings. Next we use MAML to train CNN and LSTM based models for classification. Two classification methods are examined: using MAML to classify language and using MAML to classify language and sentiment. When MAML is only use to classify language we utilized a CNN or LSTM model trained on the detected language for sentiment labelling.

In the first section we will discuss motivation and prior work. In the second section we will discuss the datasets used and how the annotations used as ground truth were derived. We will also discuss some prior methods used on these datasets as well as how our work differs from this. In the third section we will discuss the primary components of our framework. In the fourth section we will describe our experiment setup as well as an evaluation of our model performance. Finally we will look at possible improvements and future work.

1.1 Motivation

Aside from the general challenges posed by performing text classification on small corpuses, low-resource languages face particular challenges. Many low-resource languages may not be well supported by platforms. App localizations may not exist for them. Suppose a dataset is scraped from a large variety of online sources. It is possible that in many cases the language of documents in the corpus is not obvious. While in some [18] prior work human annotators are tasked with identifying languages this is not always feasible. For online tasks it is likely impractical to ask human experts to judge which language a document belongs to. When considering training for such problems it is also not always appropriate to simply sample data from a larger, high resource language. High resource languages may have different keyboard support and different demographics of users. For example, it is possible that text in some low-resource languages is primarily found in official documents. In order to alleviate this problem we utilize tweet data from both high and low resource languages. For one of our high resource languages, Spanish, we split the dataset into different regional dialects. This will allow us to consider the special case of localized high resource languages. Thus there is a need for automated systems which are able to perform multilingual text classification.

One avenue for this is to first identify the language and then use a classifier trained specifically for tasks on that language. Another method is to train a classifier which is able to identify the language as well as the sentiment. In this latter case the label set then is the cross product of the language set and the sentiment labels. We will attempt both methods. We are broadly motivated by the problem of performing natural language understanding tasks in a more online setting for low-resource languages. In the online setting we are often constrained to the few shot N-way setting. Here we are given only few examples from N classes and must quickly learn to classify. Model agnostic machine learning (MAML) [9] is a method for approaching these problems which has gained popularity. We propose utilizing MAML in order to develop a model for N-way 1-shot learning. MAML will be used to train a model for both language recognition and language sentiment recognition. The performance between these two will be compared for determining the best overall method. We will use prior work to augment our framework.

We considered training the MAML framework to classify sentiment labels. In this case we would be utilizing data from all languages, including high resource languages, for multilingual sentiment analysis. Ultimately we did not pursue this due to the extreme skew in dataset size. We also do not believe that such a classifier is appropriate for the few-shot setting we are interested in. We utilize sentiment analysis as an example text classification task. Our hope is to develop a framework which may be finetuned for different tasks.

1.2 Related work

Significant prior work has also tackled the problem of low resource languages. Broadly these fall into three categories: utilizng pre-trained language models, data augmentation, and borrowing methods from other few-shot settings.

Many of the models used to solve multilingual sentiment analysis for AfriSenti [18] utilize pre-trained language model based methods. Pretrained language models such as AfriBERTa-Large [16] aim to develop large language models from small corpuses. Training data here is compiled from BBC News localized to different African languages as well as data from Common Crawl for a subset of supported languages. AfriBERTa is able to cover eleven AFrican languages in total. The training corpus is not large, only 108.8 million tokens are used. They do not utilize transfer learning and jointly train a multilingual model from scratch. Afro-xlmr takes a different approach, instead finetuning on existing pretrained language models. The overall approach is multilingual adaptive finetuning. XLM-r is used as a base model. Their experimental results show that for classification tasks on new languages utilizing pretrained multilingual models may be an effective approach. Methods relying on these pre-trained models are able to achieve high performance on multilingual sentiment analysis with some models achieving a F1 score of 71.2 on AfriSenti.

Some prior work examines transferring PLM knowledge for programming languages. Here languages are highly structured and semantics are often known ahead of time or even completley formally defined. An insight of [6] is that in this setting monolingual PLMs are often more efficient than multilingual PLMs though they empirically may have similar performances. This study is particularly interesting in that it utilizes MTurk crowdsourcing to measure the PLM performance for code summarization. Other work [22] develops a BERT based PLM specifically for low resource languages and demonstrates strong performance on downstream tasks such as named entity recognition and part of speech tagging. This work also contributes new datasets and techniques which are particular to this language. Prior work on this particular language focused on CNN-BiLSTM based methods which we also follow.

Data augmentation is also a common method for overcoming problems with low resource languages. In [13] several low resource languages are considered simultaneously for named entity recognition. Datasets are generated by translating the corpus of a high resource language. The translation utilizes labeled sequence translation. In this model named entities are replaced with placeholders before translation and the placeholders are replaced with translated entity names after the sentence is fully translated. A monolingual data augmentation algorithm from [8] is used to diversify the datasets of each language. In the Question-Answer setting of [5] there is a particular focus on improving the multilingual embeddings. Here the problem is somewhat reversed. Other languages, including low resource languages, are used to augment an English corpus. The method here is still focused on machine translation. This method is able to improve the performance of a zero-shot English model. In [17] the multilingual mBERT model is finetuned to support multilingual documents. This improves the performance of mBERT for multilingual tasks as it is better able to handle code switching. Prior work also utilizing tweet data [3] first pre-trains an English model and then uses machine translation to adapt the model to non-English languages. The languages selected are not low resource however the work still demonstrates strong results on smaller data corpuses. The work of [19] introduces a cross lingual data augmentation framework. Here segments of input text are replaced with translations into another language, creating bilingual documents. This method is shown to give stronger performance than creating ensembles of monolingual documents in multiple languages. The researchers also demonstrate robustness against some poor quality machine translations.

There is substantial prior work on few-shot learning in general. For one-shot image recognition Siamese networks [12] are a popular method. Here a network is trained to classify whether image pairs are in the same or different classes. This model is then given one example per new class pairwise with a test image. The probability scores are used to give the one-shot prediction. The meta-learning framework of [20] is similar to the MAML framework of [9] for few-shot learning. This approach is suffer less from overfitting however the focus is on transfer learning. This is a plausible alternative approach to MAML which may be utilized in future work. A survey of few shot work in general is presented in [21].

2 Datasets

2.1 AfriSenti 2023

AfriSenti SemEval [18] is a dataset developed with fourteen African languages. Nigerian based languages have more data and so in prior work typically have higher performance models. All languages in the dataset are considered to be low-resource. Some languages, like Nigerian pidgin, may be considered multilingual. Tweets in these corpuses will contain words from multiple languages. Most work on this dataset utilizes pre-trained language models. In particular BERT, AfriBERTa-Large, and Afro-xlmr-large are frequently used.

Our work differs in that we are concerned primarily with the 1-shot case. We aimed to consider the case in which a system must classify the sentiment of unlabeled languages. Thus our model does not consider a particular language family and cannot explicitly rely on cross language similarities for training. We do similarly utilize a pre-trained transformer based model, LASER, for some of our text embedding experiments.

2.2 TASS 2019

The TASS 2019 [7] is similar to the AfriSenti dataset. Spanish is the second most commonly used language on Twitter. Tweets from multiple Spanish speaking countries as collected and manually annotated. Not all tweets are exclusively in Spanish, some contain English text as well. The TASS 2019 workshop intended for each dialect to be treated as a different language for multilingual tasks. Here, more so than in AfriSenti, there are semantic similarities between each language which can be leveraged. Prior work for this task focuses on a mixture of more traditional methods and transformer based methods. Instead past methods tend to use methods such as data augmentation and traditional architectures such as LSTMs. The remaining work focuses on using pre-trained BERT models or training new transformer based models. Performance between these two categories of methods does not tend to vary widely. Across prior work from 2019 the multilingual task F1 varies from 0.448 to 0.514. The most performant model focused on a data augmentation method. Here a translation model was first used in order to augment the datasets and next existing tweets were split in half and combined in different permutations.

As in AfriSenti our work focuses on the 1-shot case which is not studied here. We do use LSTM and pre-trained transformer models like some prior work. The composition of these methods is novel to our model.

2.3 Sentiment140

For our large, high resource corpus we utilize the annotated English tweet dataset from Sentiment140 [10]. In order to annotate the large corpus a distantly supervised method is utilized. Emoticons in the tweets are used as noisy labels for the classification process. Classic methods are used here such as SVM, Naive Bayes, and a keyword based approach. Emoticons are removed during training, a preprocessing step we also apply. The classifier is able to achieve 83% accuracy for tweet annotation. We find that this is sufficient for our purposes. This is the highest quality large English tweet dataset with annotations for positive, negative, and neutral sentiment. Future work [4] modestly improved on this performance using a Naive Bayes approach, achieving 84% accuracy. Future work also uses Multinomial Naive-Bayes for cross domain sentiment analysis. Here Sentiment140 is used as part of an ensemble of corpuses. Other work [11] utilizes a lexical database to obtain semantic information about words in the tweets.

Our model does not follow these methods, instead utilizing more recently favored approaches. These classic NLP methods may be promising however we do not evaluate them against our CNN and LSTM models. This is in part due to our usage of MAML which utilizes the weight tensors of different layers in neural architectures.

3 Model

3.1 Classifier

We utilize two types of classifier: a CNN based model and a bidirectional LSTM based model. We achieve comparable performance in different settings with both models.

CNN

We use CNN as a classifier because of its ability to effectively extract features from the input embeddings and use the features to identify patterns that are associated with each label. Multiple convolutional layers process information at different levels of abstraction. The combination of multiple convolutional layer and maxpooling layers are able to extract both local and global context.

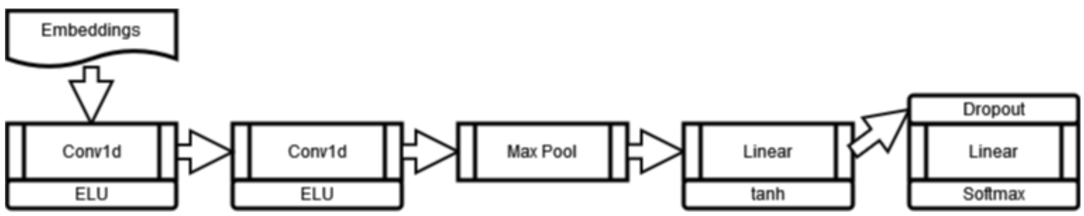


Figure 1: Architecture for CNN sentiment classifier.

biLSTM The Bidirectional Long Short-Term Memory (BiLSTM) model is particularly adept at processing sequential data. The core idea of LSTM is to remember significant information over long sequences, effectively addressing the vanishing gradient problem that hampers standard RNNs. In the context of sentiment analysis, LSTM models can discern the contextual relevance and semantic relationships in textual data, crucial for determining the sentiment expressed. The model's ability to maintain information over lengthy sequences allows it to understand the overall sentiment of a text, which may be influenced by both the immediate context and more distant elements in the sequence. The Bidirectional aspect of the LSTM is a further enhancement, allowing the network to process information in both forward and backward directions. In practical terms, this means the BiLSTM can capture context from both the past and the future within a sequence. This design allows each layer to have two sets of hidden states, effectively doubling the amount of information captured about each sequence. The model uses multiple LSTM layers, which can enhance its ability to model complex relationships in the data. The dropout layers introduced in the model help mitigate overfitting, a common problem in MAML. The final fully connected layers serve to map the learned features to the desired output dimension, which can vary from sentiment to the joint sentiment-language labels.

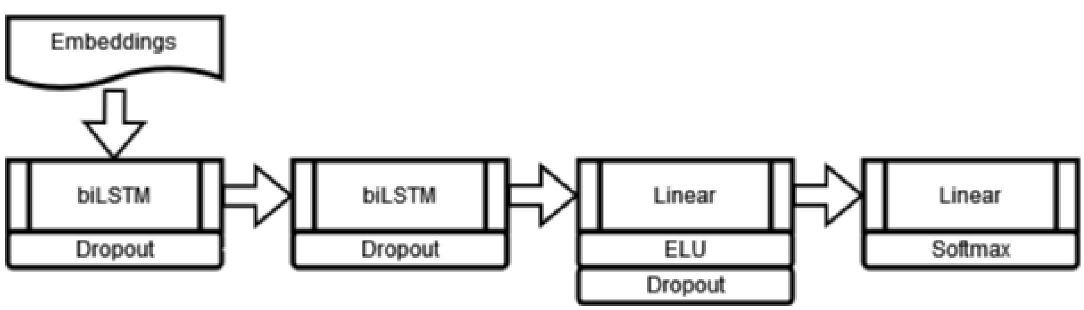


Figure 2: Architecture for LSTM sentiment classifier.

3.2 LASER

LASER (Language-Agnostic SEntence Representations) is a multilingual sentence representation that works with more than 90 languages and 28 alphabets.[2]. LASER is distinctive because of its ability of embedding texts in different languages into the same vector space. It uses a single model to handle different types of languages and had been used to perform one-shot transfer of models from higher resource languages to low resource languages.

We used the LASER-2 model which covers 93 different languages to generate embedding of texts to use as input to MAML, which is covered later in this section. We hope the language agnostic feature of LASER embedding would have improved performance compared to FastText embeddings, for which each language has its own model.

3.3 SuperGen

Super Gen[15] uses a unidirectional language model to generate training data, guided by label-descriptive prompts. Here, we have replicated a similar approach using the GPT-2 model from the transformers library for English and the gpt-2-small-spanish model for Spanish. These models are utilized to generate training data when supplied with label-descriptive prompts created by a unidirectional model. However, despite rigorous hyperparameter tuning, the text output from the GPT-2 model has not met expectations, decreasing performance in our experiments. This issue has also been acknowledged on the Hugging Face website. Our outcomes with Super Gen could have been improved by utilizing GPT-3 for text generation, which addresses the challenges encountered with GPT-2. Specifically, GPT-2 tends to produce similar responses for nearly identical prompts and suffers from lower quality text generation, issues that GPT-3 effectively overcomes. We were limited by what we could generate from multilingual unidirectional models. One method which may overcome this issue is utilizing machine translation.

3.4 MAML

Meta-learning is a method of machine learning in which an algorithm is used to learn how to adapt to new tasks quickly. The model agnostic (MAML) method of [9] considers some model f which learns a set of parameters θ. The model is trained to be able to complete some possibly infinite set of tasks $\mathcal{T}$. The model is trained on a subset of the tasks. For each task $\mathcal{T}_i$ which is seen after training, a new parameter set θ_i' is learned in a few shots to perform $\mathcal{T}_i$. Each task has a different loss function associated with it, $\mathcal{L}_{\mathcal{T}_i}$. So for each θ_i', we have that

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{1}$$

for a single gradient update where the step size α is a tunable parameter. The parameter θ for the entire MAML algorithm then is based on a linear update of all the task losses. The update is then performed by

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \tag{2}$$

where β is the step-size for the overall meta algorithm and $p(\mathcal{T})$ is the distribution over the task set. The distribution is important when we suppose that the task likelihood is not uniform and have some prior over the likelihood of encountering different tasks. Here we consider the task distribution as being equally likely. This way the high resource languages are not over sampled. The parameter update does not weight tasks by their likelihood. Even tasks which are not encountered contribute equally to the update of the parameters.

We consider the 5-Way 1-Shot usage of MAML. We sample five labels at a time and are given one training example. This training sample is used to predict the class labels of fifteen query samples. The loss incurred from the query sample training is used to update the model parameters. In the language only case we take one tweet embedding from each the five randomly selected languages. In the language and sentiment case we take one tweet embedding from each of the five language-sentiment pairs. After some tuning we selected Due to time constraints we were not able to further tune the hyperparameters or attempt significantly longer training times. As MAML is known to require very long training times we believe that a combination of more tuning and longer training will be able to improve performance.

We adapt the implementation of [14] for our MAML implementation. The original code is for image recognition which we modify for text classification. We additionally add functionality for utilizing LSTM layers. Though not directly comparable, their performance on 5-way 1-shot image recognition is 46.2%. This is slightly below the 48.7% reported in [9] using the same model. The difference may be due to different hyperparameter tuning and implementation details.

4 Experiments and evaluation

4.1 Preprocessing

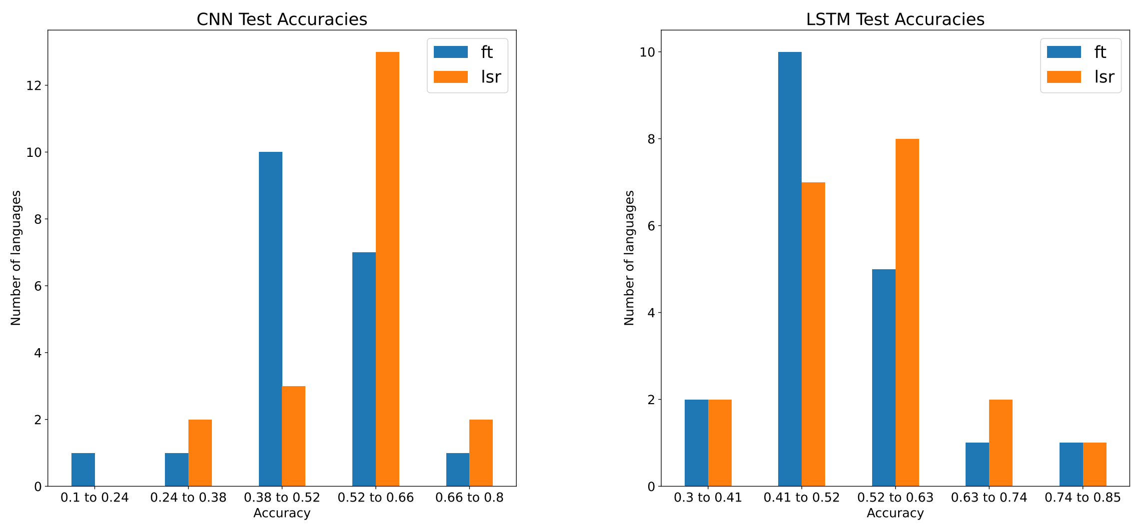
We apply a language agnostic pre-processing method. In the TASS 2019 dataset we remove tweets which are annotated as having no sentiment. Next, across all languages we remove URLs, emails, and phone numbers. These links are generally not informative for sentiment. Next we remove all punctuation and extra whitespace. We then pass the data to one of two embedding models.

4.2 Embeddings

We utilize two text embedding models, FastText and LASER. FastText produces lower dimensional embeddings of size 100 and LASER produces higher dimensional embeddings of size 1024. To train a FastText model we utilize our training corpus with word2vec. Our hope is that the usage of subwords will allow us to capture meaningful embeddings from our small corpus.

4.3 Model training

We train the model utilizing a P100 GPU. Training times for MAML are very long. Using the LASER embeddings they can take up to twelve hours. Using larger datasets and more training epochs would exceed the maximum compute time available to us. Using the FastText embeddings however, training generally completes within two hours, a relatively quick training time. We were unable to perform extensive hyper-parameter tuning due to the long training times. We validate that the CNN and LSTM architectures are appropriate by running training them in a standard sentiment analysis task. Here performance ranges with the availability of data but is within a reasonable range. These results may be found in Figure 3.

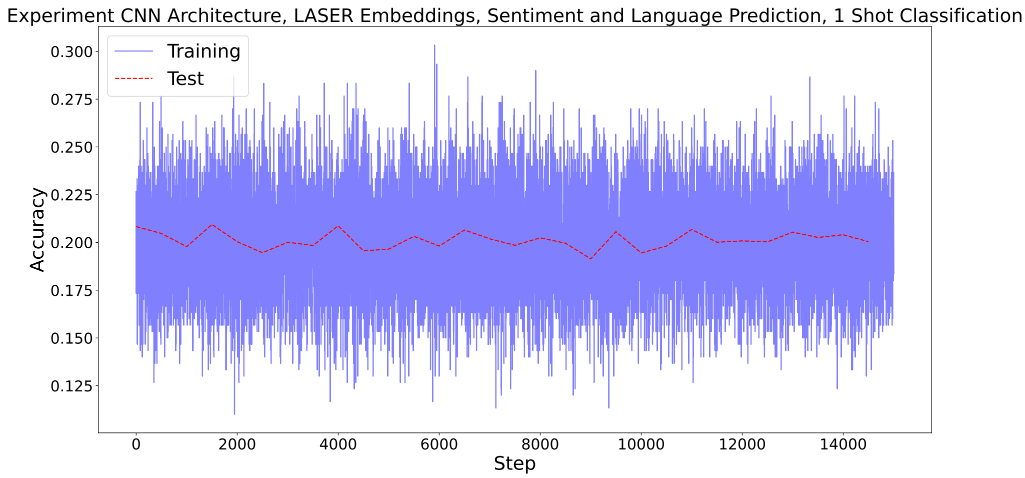


(a) CNN accuracies.

(b) CNN accuracies.

Figure 3: CNN and LSTM training accuracies.

We also attempted to improve our performance by utilizing LASER embeddings. Our hope was that because the embeddings are built from larger pre-trained models that they would give stronger results in the MAML step. We did not find that this was the case, indicating that the cause of low performance lies within the MAML architecture. We show the performance using LASER embeddings below in Figure 4.



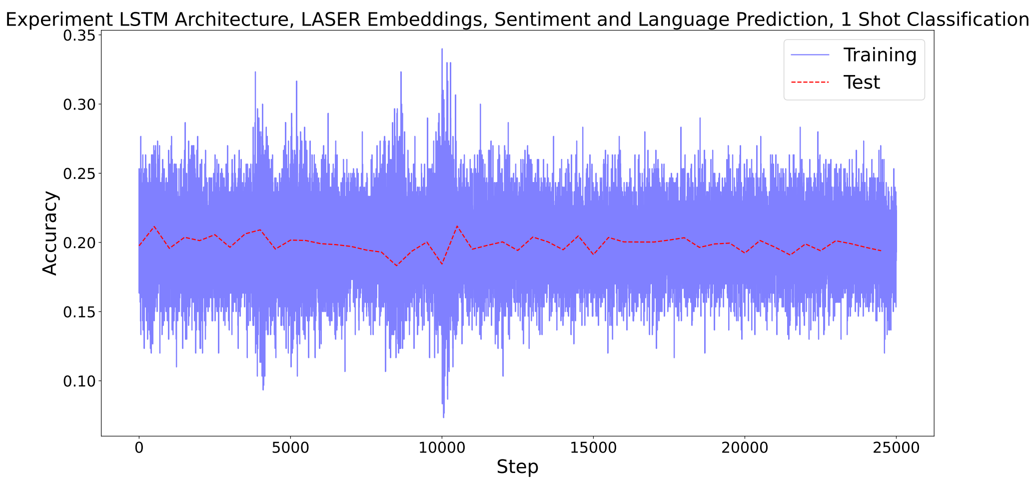


Figure 4: Classifying with LASER embeddings and CNN/LSTM architecture. Predictions on sentiment and language joint labels.

4.4 Ablations

No LASER

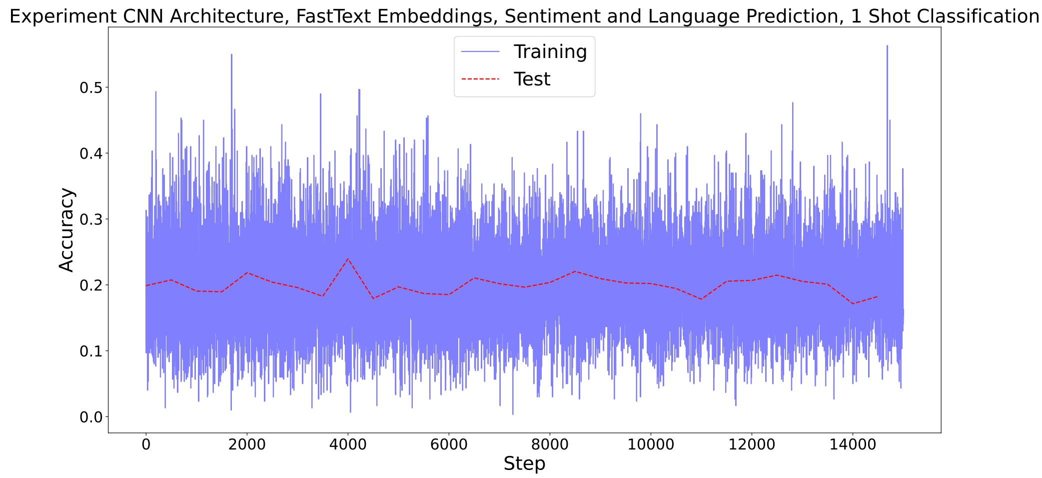
We demonstrate the degradation of performance by not using LASER embeddings. Instead, we will use lower dimension FastText embeddings trained on our collected corpus of tweets. The performance of utilizing MAML with FastText embeddings is given below in Table 1.

	Predict language and sentiment	Predict language
CNN	0.2015	0.2025
LSTM	0.210	0.2019

Table 1: Performance of MAML on FastText embedded tweets, hold out test set accuracies. Average accuracy is shown.

Performance here is slightly better than random guess. There is substantial variation between the test and train accuracies in many cases. During training language prediction with CNN can attain 0.41 accuracy and 0.347 with

LSTM. For prediction with sentiment label and language, CNN is able to attain 0.387 accuracy and LSTM is able to attain 0.31 accuracy. This substantial gap suggests that the MAML model is overfitting. This problem may be alleviated with data augmentation or by improved hyperparameter tuning. MAML is known to suffer from overfitting and subsequent work [1] has suggested methods for reducing overfitting and improving training performance. Due to time constraints we are unable to evaluate these changes. Performances may be found in Tables 5 and 6. These problems can also be seen in the wide variation in training accuracies across batches.



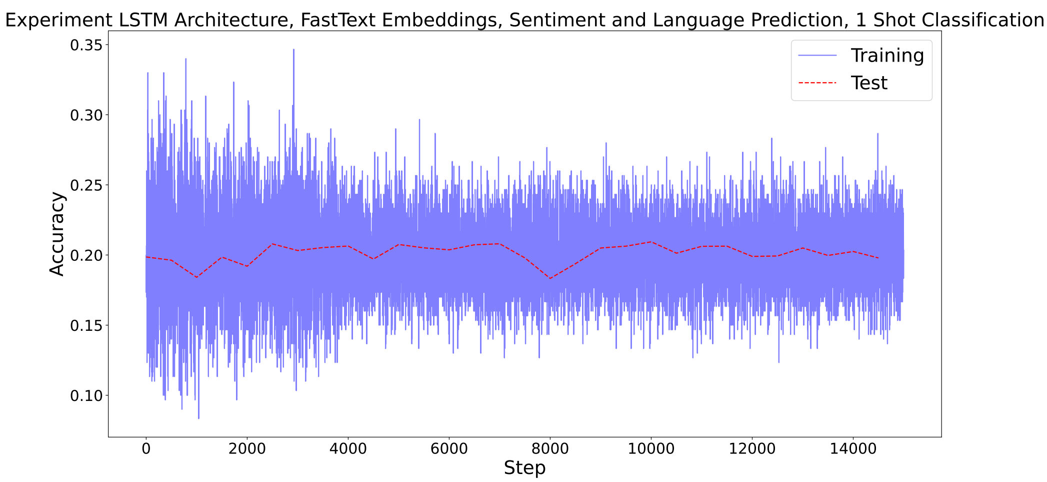
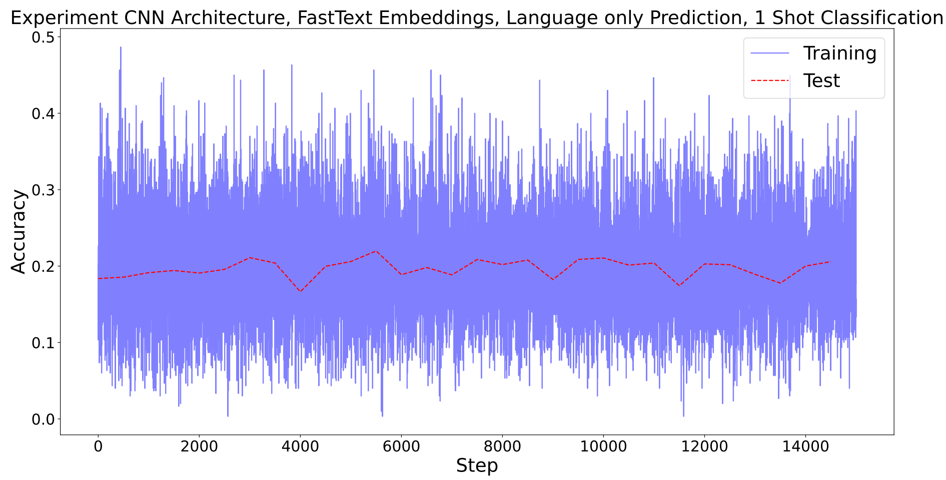


Figure 5: Classifying with FastText embeddings and CNN/LSTM architecture. Predictions on sentiment and language. Performance between the two architectures are comparable.

5 Conclusion

5.1 Future work

There are many further improvements which could be made to increase the performance of this model. On a more basic level, the CNN and LSTM architectures could be continuously tuned for better performance. The hyperparameters for the MAML model could similarly be tuned. More training epochs could also be used to improve performance. We will also attempt to incorporate some of the improvements to MAML suggested in [1]. It may also



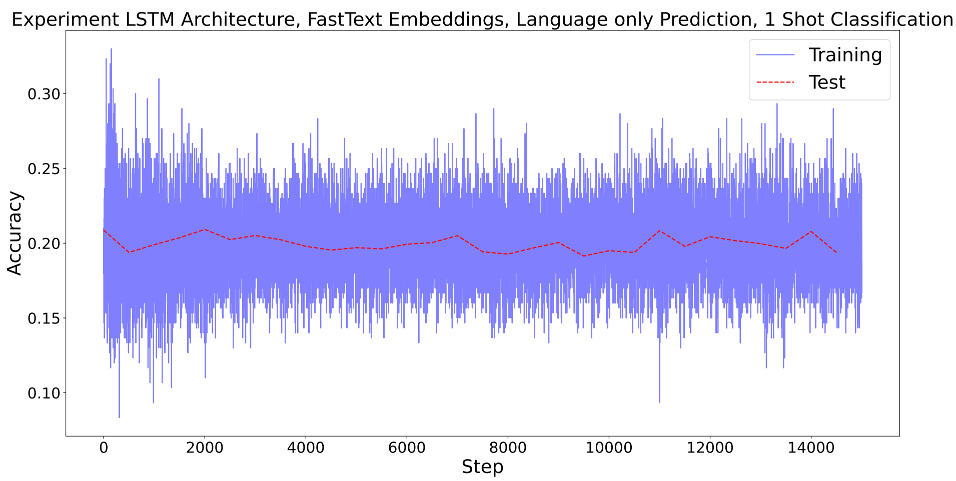


Figure 6: Classifying with FastText embeddings and CNN/LSTM architecture. Predictions on only language. Reducing the number of classes to predict does not increase performance.

be effective to try the MTL approach of [20]. For data augmentation we may also try the translation based methods used in prior work.

References

[1] Antoniou, A., Edwards, H., and Storkey, A. How to train your maml. *arXiv preprint arXiv:1810.09502* (2018).

[2] Artetxe, M., and Schwenk, H. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics 7* (Nov. 2019), 597–610.

[3] Barriere, V., and Balahur, A. Improving sentiment analysis over non-english tweets using multilingual transformers and automatic translation for data-augmentation. *arXiv preprint arXiv:2010.03486* (2020).

[4] Behera, R. N., Roy, M., and Dash, S. A novel machine learning approach for classification of emotion and polarity in sentiment140 dataset. In *Conference: 3rd International Conference on Business & Information Management* (2016).

[5] Bornea, M., Pan, L., Rosenthal, S., Florian, R., and Sil, A. Multilingual transfer learning for qa using translation as data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 12583–12591.

[6] Chen, F., Fard, F. H., Lo, D., and Bryksin, T. On the transferability of pre-trained language models for low-resource programming languages. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension* (2022), pp. 401–412.

[7] Díaz-Galiano, M. C., Vega, M. G., Casasola, E., Chiruzzo, L., Cumbreras, M. Á. G., Martínez-Cámara, E., Moctezuma, D., Ráez, A. M., Cabezudo, M. A. S., Tellez, E. S., Graff, M., and Miranda-Jiménez, S. Overview of tass 2019: One more further for the global spanish sentiment analysis corpus. In *IberLEF@SEPLN* (2019).

[8] Ding, B., Liu, L., Bing, L., Kruengkrai, C., Nguyen, T. H., Joty, S., Si, L., and Miao, C. Daga: Data augmentation with a generation approach for low-resource tagging tasks. *arXiv preprint arXiv:2011.01549* (2020).

[9] Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning* (2017), PMLR, pp. 1126–1135.

[10] Go, A., Bhayani, R., and Huang, L. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford 1*, 12 (2009), 2009.

[11] Goel, A., Gautam, J., and Kumar, S. Real time sentiment analysis of tweets using naive bayes. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)* (2016), IEEE, pp. 257–261.

[12] Koch, G., Zemel, R., Salakhutdinov, R., et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop* (2015), vol. 2, Lille.

[13] Liu, L., Ding, B., Bing, L., Joty, S., Si, L., and Miao, C. Mulda: A multilingual data augmentation framework for low-resource cross-lingual ner. 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)* (2021), pp. 5834–5846.

[14] Long, L. Maml-pytorch implementation. `https://github.com/dragen1860/MAML-Pytorch`, 2018.

[15] Meng, Y., Huang, J., Zhang, Y., and Han, J. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems 35* (2022), 462–477.

[16] Ogueji, K., Zhu, Y., and Lin, J. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning* (Punta Cana, Dominican Republic, Nov. 2021), Association for Computational Linguistics, pp. 116–126.

[17] Qin, L., Ni, M., Zhang, Y., and Che, W. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp. *arXiv preprint arXiv:2006.06402* (2020).

[18] Salahudeen, S. A., Lawan, F. I., Wali, A., Imam, A. A., Shuaibu, A. R., Yusuf, A., Rabiu, N. B., Bello, M., Adamu, S. U., and Aliyu, S. M. HausaNLP at SemEval-2023 task 12: Leveraging African low resource TweetData for sentiment analysis. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)* (Toronto, Canada, July 2023), Association for Computational Linguistics, pp. 50–57.

[19] Singh, J., McCann, B., Keskar, N. S., Xiong, C., and Socher, R. Xlda: Cross-lingual data augmentation for natural language inference and question answering. *arXiv preprint arXiv:1905.11471* (2019).

[20] Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 403–412.

[21] Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur) 53*, 3 (2020), 1–34.

[22] Yohannes, H. M., and Amagasa, T. Named-entity recognition for a low-resource language using pre-trained language model. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (2022), pp. 837–844.

A Acknowledgements

The authors would like to thank Prof. Jiawei Han and the TAs for their feedback which has strengthened the work. Some text from the original proposal is adapted for use here.

B Code

The training code is available at `https://github.com/andyclee/meta-sa`. Training data is not included here for size limitations but is available upon request.

C Additional data and evaluations

Below we show our training and meta-test results.

	Predict language and sentiment	Predict language
CNN	0.230	0.220
LSTM	0.203	0.212

Table 2: Performance of MAML on FastText embedded tweets, meta-test accuracies. Best performances are shown.

	Predict language and sentiment	Predict language
CNN	0.387	0.41
LSTM	0.347	0.31

Table 3: Performance of MAML on FastText embedded tweets, meta-train accuracies. Best performances are shown.

We also wished to see if a direct modification to 5-shot learning would improve our accuracies. We found that this was not the case and that any issue underlying the 1-shot case also affect the 5-shot case.

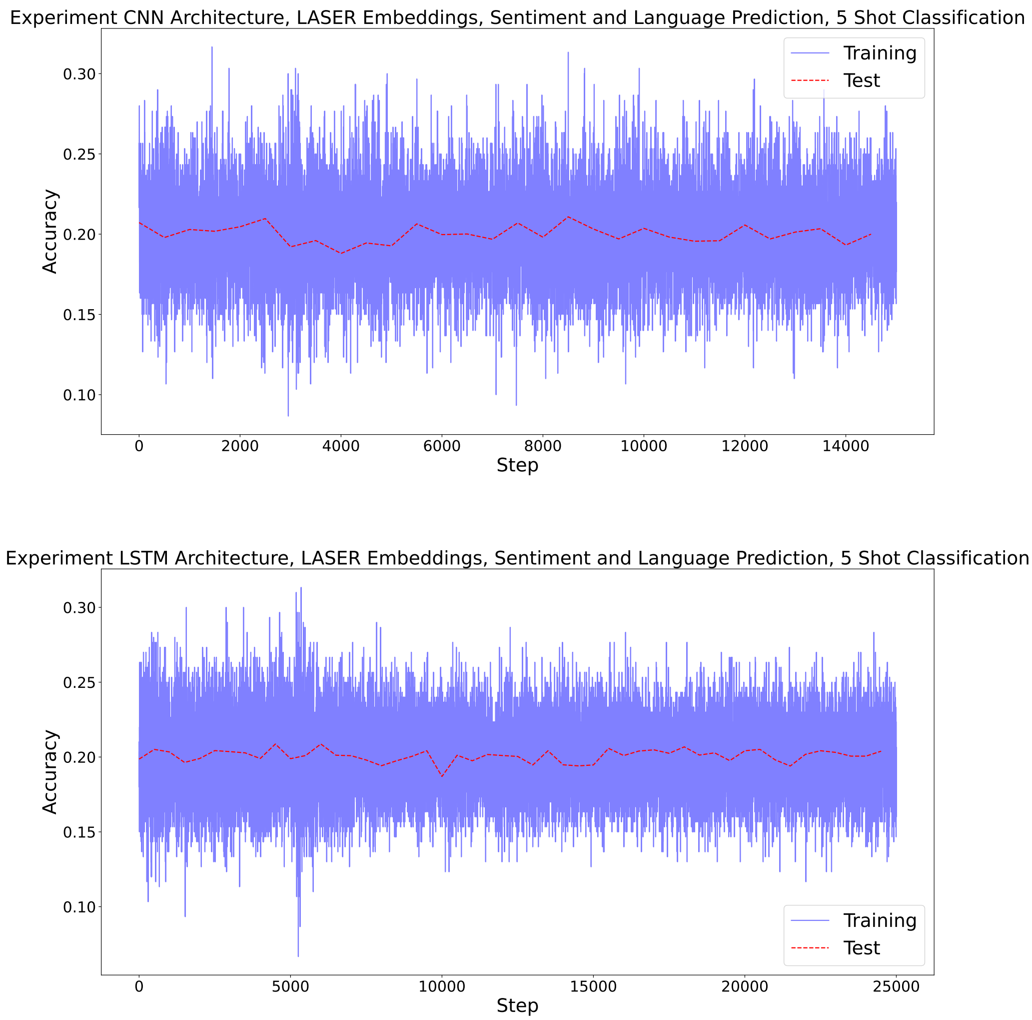


Figure 7: Training results for 5-shot learning. The LASER embeddings are used here for both architectures. Classification is done on the join sentiment, language label set.