Unsupervised Self-Training for Unsupervised Cross-Lingual Transfer

Akshat Gupta[1], Sargam Menghani[1], Sai Krishna Rallabandi[2], Alan W Black[2]
[1]Department of Electrical and Computer Engineering, Carnegie Mellon University
[2]Language Technologies, Institute, Carnegie Mellon University
{akshatgu, smenghan}@andrew.cmu.edu, {srallaba, awb}@cs.cmu.edu

Abstract

Labelled data is scarce, especially for low resource-languages. This beckons the need to come up with unsupervised methods for natural language processing tasks. In this paper, we introduce a general framework called Unsupervised Self-Training, capable of unsupervised cross-lingual transfer. We apply our proposed framework to a two-class sentiment analysis problem of code-switched data. We use the power of pre-trained BERT models for initialization and fine-tune them in an unsupervised manner, only using pseudo labels produced by zero-shot predictions. We test our algorithm on multiple code-switched languages. Our unsupervised models compete well with their supervised counterparts, with their performance reaching within 1-7% (weighted F1 scores) when compared to supervised models trained for a two class problem.

1 Introduction: Unsupervised Self-Training

Our proposed algorithm is centred around the idea of creating an unsupervised learning algorithm, which is able to harness the power of cross-lingual transfer in the most efficient way possible, with the aim of making unsupervised predictions. In its most fundamental form, our proposed Unsupervised Self-Training (UST) algorithm [1] is shown in Figure 1.

We begin by producing zero-shot results for sentiment classification using a selected pre-trained model trained for the same task. We use a RoBERTa (Liu et al., 2019; Devlin et al., 2018) based sentiment classification model (Barbieri et al., 2020) trained on an English Twitter Corpus (Rosenthal et al., 2017). We refer to this model as our *base model*. From the predictions made, we select the top-N most confident predictions made by the model. The confidence level is judged by the final softmax scores. Making the zero-shot predictions and selecting sentences make up the *Initialization block* as shown in Figure 1.

We then use the pseduo-labels predicted by the zero-shot model to fine tune our base model. This is done in the *Fine-Tune Block*. We remove the previously selected sentences used to fine-tune the base model from our dataset. After that, predictions are made on the remaining dataset with the unsupervised fine-tuned model. This is done in the *Prediction Block*. We again select sentences based on their softmax scores in the *Selection Block* for fine-tuning the model in the next iteration. These steps are repeated until we've gone through the entire dataset or until a stopping condition. At all fine-tuning steps, we only use the predicted pseudo-labels as ground truth to train the model, which makes the algorithm completely unsupervised.

As the first set of predicted pseudo-labels are produced by a zero-shot model, our framework is very sensitive to initialization. Care must be taken to initialize the algorithm with a *compatible model*. For example, for the task of sentiment classification of Hinglish Twitter data, an example of a compatible initial model would be a sentiment classification model trained on either English or Hindi sentiment data. It would be even more compatible if the model was trained on Twitter sentiment data, the data thus being from the same domain.

Additionally, to improve performance, we can use several training strategies in the Selection Block. Instead of selecting the top-N predicted samples, we can select the top-N samples for each predicted class. This selection strategy is called **Vanilla Selection**. This strategy works well when the underlying distribution of data is balanced. In case of an unbalanced data distribution, we see improved results when top N_i samples are selected from each class i in the dataset. This needs a rough approximation of the underlying data distribution.

[1] The code for the framework can be found here: https://github.com/akshat57/Unsupervised-Self-Training

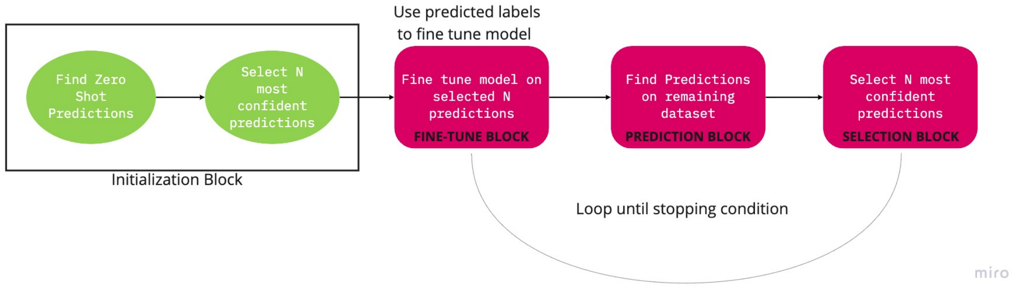
Use predicted labels
to fine tune model
Find Zero Shot Predictions
Select N most confident predictions
Initialization Block
Fine tune model on selected N predictions
FINE-TUNE BLOCK
Find Predictions on remaining dataset
PREDICTION BLOCK
Select N most confident predictions
SELECTION BLOCK
Loop until stopping condition
miro

Figure 1: A visual representation of our proposed Unsupervised Self-Training framework.

TRAIN LANGUAGE	ZERO-SHOT		UST VANILLA		UST RATIO		SUPERVISED	
	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy
Hinglish	0.32	0.36	0.84	0.84	0.84	0.84	0.91	0.91
Spanglish	0.31	0.32	0.77	0.76	0.77	0.77	0.78	0.79
Tanglish	0.15	0.16	0.68	0.63	0.79	0.80	0.83	0.85
Malayalam – English	0.17	0.14	0.73	0.71	0.83	0.85	0.90	0.90

Table 1: Performance of best Unsupervised Self-Training models for Vanilla and Ratio selection strategies when compared to performance of supervised models. The F1 scores represent the weighted average F1.

This selection strategy is called **Ratio Selection**. Knowing optimal selection strategies are vital to achieving the optimum performance.

2 Datasets

We test our proposed framework on four different languages - Hinglish (Patwa et al., 2020), Spanglish (Patwa et al., 2020), Tanglish (Chakravarthi et al., 2020b) and Malayalam-English (Chakravarthi et al., 2020a). We gently request the reader to refer to the respective papers for more details. The choice of datasets, apart from covering three language families, have two other notable charactersitics. The Hinglish and Spanglish datasets are roughly balanced, while the other two are highly imbalanced. The chosen datasets are also from two different domains - the Hinglish and Spanglish datasets are a collection of Tweets whereas Tanglish and Malaylam-English are a collection of Youtube Comments. Each of the four code-mixed datasets selected are written in the latin script. Thus our choice of datasets does not take into account mixing of different scripts.

3 Results

The results of our experiments are presented in Table 1. The zero-shot results correspond to the zero-shot prediction accuracy of the base model. The base model is trained on a Twitter corpus, thus we see better zero-shot results for Hinglish and Spanglish. Using the Vanilla Selection Strategy and fine tuning the model under the UST framework significantly improves the zero-shot results. Using the Ratio Selection Strategy significantly improves the performance of the Tanglish and Malayalam-English dataset as they have the highest amount of data imbalance. The imbalance is minor in Hinglish and Spanglish. We see the Ratio Selection results are very close the supervised results, which essentially provide a ceiling for the unsupervised algorithm. The supervised models are binary counterparts of (Gupta et al., 2021b). An extensive analysis of the learning dynamics of the algorithm shows that with every iteration, the performance of the fine-tuned model improves for sentences with larger amounts of code-mixing. Eventually, the model begins to understand the code-mixed data.

4 Conclusion

We propose the Unsupervised Self-Training framework[2] and show results for unsupervised sentiment classification of Code-Switched data. The algorithm is comprehensively tested for four very different code-mixed languages - Hinglish, Spanglish, Tanglish and Malayalam-English, covering many variations including differences in language families, domains, dataset sizes and dataset imbalances. The unsupervised models performed competitively when compared to supervised models. We also present training strategies to optimize the performance of our proposed framework.

[2] The complete work can be found here: (Gupta et al., 2021a)

References

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P McCrae. 2020a. A sentiment analysis dataset for code-mixed malayalam-english. *arXiv preprint arXiv:2006.00210*.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John P McCrae. 2020b. Corpus creation for sentiment analysis in code-mixed tamil-english text. *arXiv preprint arXiv:2006.00206*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Akshat Gupta, Sargam Menghani, Sai Krishna Rallabandi, and Alan W Black. 2021a. Unsupervised self-training for sentiment analysis of code-switched data. *arXiv preprint arXiv:2103.14797*.

Akshat Gupta, Sai Krishna Rallabandi, and Alan Black. 2021b. Task-specific pre-training and cross lingual transfer for code-switched data. *arXiv preprint arXiv:2102.12407*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. *arXiv e-prints*, pages arXiv–2008.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518.