Exploration of Emotion Classification for Low-Resource Languages with XLM-RoBERTa

Isaac Schifferer
isaacjs2@illinois.edu

Abstract

Emotion classification, like all tasks in computational linguistics, becomes more difficult when dealing with low-resource languages. Transformer-based general language models are one potential solution for bridging the gap in performance between high- and low-resource languages. This paper explores the application of one such model, XLM-RoBERTa (XLM-R) to the task of emotion classification. To achieve this, XLM-R is fine tuned on groups of languages that share a common feature. Each model is tested on a set of high- and low-resource languages to observe which features lead to high performance in a range of cross-lingual settings. A custom model is then trained for each low-resource language with the insight gained from the initial experiments. I show that in the case of a language having little domain-specific data, a model trained on strategically selected languages can result in a significant increase in performance relative to a baseline of only training on the limited data of the low-resource language.

1 Introduction

Emotion classification is a challenging task for a number of reasons. In general, it is difficult because each person has a unique combination of life experiences and predispositions that influences how they interpret and react to the world around them. Adding to this challenge is the task of having to quantify and then classify those emotional experiences with an algorithm. Furthermore, high-quality data is difficult to obtain and costly to annotate. A solution to this challenge, the one used in this paper, is to automatically label data, but this results in having noisy data. The same task in a low-resource setting becomes even more challenging, because emotion is less well documented, and data is scarcer than ever. One approach to mitigating this additional difficulty is to utilize knowledge gained from high-resource languages via language-agnostic embeddings, which is what general language models like XLM-R try to achieve.

This task is relevant because it has many applications. For example, in the commercial world, detecting emotion in reviews of products can help companies to find out how people feel about their products at a much larger scale than they could have previously. Similarly, emotion classification can be used on social media to gauge how the general population feels about current events, which could indicate to a news organization what they should focus their coverage on. From an academic standpoint, emotion classification can be used to study how different groups of people react to certain events or feel about certain things.

2 Problem Definition

The goal of this paper is to explore approaches for emotion classification in low-resource languages. By definition, most, if not all low-resource languages do not have emotion-labeled corpora large enough to train a model in any meaningful way. So, one approach is to fill in the gaps with data from other languages that will help the model to acquire knowledge about emotion that can be applied to low-resource languages. An ideal system that carries out this task would be able to encode text from a wide range of languages in a language-agnostic manner in order to process them for their emotional content and then return the correct emotion label for each text.

Part of this problem also has to do with choosing the best set of languages to train the model on, languages that are similar to the target language in a way that is relevant to the task of emotion classification.

3 Previous Work

Emotion classification is very much an active point of research today. At SemEval-2018, one of the tasks was to build an emotion classification model, with subtasks for English, Spanish, and Arabic. The results were aggregated in Mohammad et al. (2018) and show that the top teams in the task had accuracies in the range of 45-60%. These models often employed deep learning or neural networks to achieve their scores.

Specific to XLM-R, there has been some research on its ability to do emotion classification for single languages. In a paper exploring several text classification problems for Bengali, an F1 score of 70.6 was achieved for emotion classification on a data set from YouTube (Alam et al., 2020). A model trained on social media comments in Tamil achieved an F1 score of .27 for an emotion set of 12 in Gokhale et al. (2022). Lastly, Vera et al. (2021) achieved an F1 score of 71.7 for their emotion classification model for Spanish.

Hassan et al. (2021) experimented with cross-lingual methods of emotion classification for mBERT, Google's multilingual general language model. They showed that models can be created that are only trained on one language, in this case English, that are almost as effective for other languages, in this case Arabic and Spanish, when compared to models specifically trained for those other languages.

The paper that published the dataset used in this paper had a similar approach to emotion classification for low-resource languages. Using mBERT, Lamprinidis et al. (2021) trained several models with various combinations of training data. Most notably, they trained a pair of models, one with data from only Indo-European languages, and one with data from only non-Indo-European languages. They showed that languages perform significantly better on the model with the matching language family category, with an average difference of 14.8 in macro-F1 score. They also look at zero-shot learning for low-resource languages, achieving macro-F1 scores generally between 30 and 40. Hoping to expand and improve on these efforts, this paper looks at multiple language families as well as word order and grapheme type and analyzes the results of zero-shot learning on models trained with languages stratified by feature.

4 Approach

4.1 The Universal Joy Data Set

The data set used in this paper comes from Lamprinidis et al. (2021). It contains a diverse set of 18 languages. The data itself was collected from public Facebook posts with a "feeling" tag. The 27 initial emotion tags from the posts were mapped onto five basic emotions: anger, anticipation, fear, joy, and sadness. Preprocessing was done to replace names, locations, photos, email addresses, and URLs with generic placeholders. Names were replaced with "[PERSON]," locations were replaced with "[LOCATION]," and so on. Any number was replaced with 0. The amount of data available for each language varies greatly, from 869 posts for Bengali to almost 300,000 posts for English, though for the sake of consistency, the small version of the data set is used for the largest languages (English, Portuguese, Chinese, and Tagalog), which includes 2,947 posts for each language. The average number of posts for the data sets used in this paper is 3,847.

The distribution of emotions is fairly unbalanced for these data sets. For example, the Chinese data set has 276 posts for anger, 657 for anticipation, 63 for fear, 1600 for joy, and 351 for sadness. This distribution is matched for the other data sets, with the exception of English, which has slightly more anticipation posts than joy posts.

4.2 XLM-RoBERTa

The base model for each model trained in this paper is a version of XLM-RoBERTa, the multilingual general language model from Facebook (Conneau et al., 2019). It was trained with 2.5 TB of CommonCrawl and Wikipedia data from 100 languages. The specific version used for these experiments was fetched from TensorFlow Hub and has 12 layers, a hidden size of 768, and 12 attention heads. On top of the preprocessing described in the previous section, the data was also preprocessed with a preprocessor built for XLM-R, also accessed via TensorFlow Hub. Each piece of data is encoded by XLM-R and no additional features are used.

The primary training languages used in the models–English, French, Portuguese, Chinese, Indonesian, Vietnamese, Thai–were each represented in at least 45 GB of the data that XLM-R was trained on, and so are defined high-resource lan-

guages for the sake of this paper. Conversely, Burmese, Tagalog, and Khmer are defined as low-resource languages because they each contributed less than 5 GB of data to the training of the base XLM-R model.

4.3 Experiments

Each model used in the experiments was trained on the data from one or more languages with an 80-20 train-test split. The models were also evaluated on 20 percent of each data set, the only exception to this being that if one of the languages with less than 1,000 examples was not being trained on, the whole set was used for testing. Apart from the base XLM-R model, models consisted only of the XLM-R preprocessor and a softmax activation layer. Each model used a batch size of 1, a learning rate of 1×10^{-5}, and trained for $12/\#\, languages\, trained\, on$ epochs so that each model was trained on roughly the same amount of data regardless of the number of languages it was trained on.

The evaluation metric used was a weighted F1 score. It was calculated as the mean of the F1 scores for each class, weighted by the number of true instances of each class in the test set. This was used as opposed to the standard F1 score due to the classes being very unbalanced.

Each baseline model was trained on a single high-resource language. The training languages were English, French, Portuguese, Chinese, Indonesian, Vietnamese, Thai, and the testing languages are listed in Table 1.

4.3.1 Feature Testing

For each of language family, word order, and grapheme type, a model was constructed for each feature value represented in the Universal Joy data set.

First, models were trained for each language family represented in the data. The Indo-European model was trained on English, French, and Portuguese. The Sino-Tibetan model was trained on Chinese and Burmese. The Austronesian model was trained on Indonesian, Tagalog, and Malay. The Austroasiatic model was trained on Khmer and Vietnamese. Finally, the Tai-Kadai model was trained on Thai.

Next, languages were stratified by most common word order in sentences. The Subject-Verb-Object (SVO) model was trained on English, Chinese, and Indonesian, the Subject-Object-Verb (SOV) model was trained on Bengali, Burmese, and Dutch, and the Verb-Subject-Object (VSO) model was trained on Tagalog.

Lastly, models were trained based on grapheme type, whether the most basic meaningful unit in the language's writing system is letters or syllables, or full words. The alphabetic model was trained on French, Indonesian, and Vietnamese. The syllabic model was trained on Hindi and Thai. The word-based model was trained on Chinese.

In the case of the Tai-Kadai language family model and the word-based grapheme model, which were trained only on Thai and Chinese respectively, a new model was not trained because they would be identical to the corresponding baseline models.

4.3.2 Custom Models for Low-Resource Languages

After results were gathered for the feature testing stage, a custom model was trained for each low-resource language. Each one was trained on data from languages that share the feature values with the low-resource language that led to the best relative performance in the feature testing stage. The models were trained on the two languages that best fit these criteria, with tiebreakers going to languages with more available data.

For example, Burmese is a Sino-Tibetan language with SOV word order and a syllabic writing system. In the feature testing stage, it performed the best on the Indo-European, SOV, and syllabic models compared to the others in their respective categories. So, the custom model for Burmese would ideally be trained on two languages with SOV word order and a syllabic writing system.

The Burmese model ended up being trained on Hindi, because it has SOV word order and a syllabic writing system, and Thai, because it is syllabic. The Tagalog model was trained on Indonesian and Malay for being Austronesian languages with alphabetic writing systems. Khmer only performed the best on the model that matched its feature value for word order, so its model was trained on French and Indonesian for having SVO word order. These results were compared to models trained only on the corresponding low-resource language, in the same manner as the other baseline models.

ISO Code	Language
en	English
fr	French
pt	Portuguese
zh	Chinese
id	Indonesian
vi	Vietnamese
th	Thai
my	Burmese
tl	Tagalog
km	Khmer

Table 1: Testing languages

	en	fr	pt	zh	id	vi	th
en	57.7	54.5	53.9	55.0	49.4	52.2	50.4
fr	52.6	60.4	53.4	53.0	51.9	48.9	52.3
pt	51.2	57.5	60.0	47.8	50.4	46.6	51.0
zh	56.5	56.7	57.5	66.5	55.9	52.6	58.2
id	46.2	49.7	49.4	49.9	52.8	45.8	46.4
vi	47.2	49.9	47.3	50.0	46.8	47.1	46.2
th	49.9	51.2	48.8	53.9	53.7	47.9	53.9
my	46.6	48.8	43.0	39.0	44.3	36.7	47.3
tl	42.1	46.1	42.8	38.6	43.6	38.5	38.8
km	48.1	45.6	46.9	40.2	45.0	45.0	47.2

Figure 1: Baseline Models

5 Results

Figure 1 shows the results of the baseline models. Each column represents one of the models, and the column label shows which language it was trained on. Figures 2 through 4 show the results for each set of language feature models. Table 2 shows the results of the custom-trained models for the low-resource languages.

The baseline models generally performed the best on the language it was trained on. The main exception to this was that for the Indonesian, Vietnamese, and Thai models, the Chinese test set yielded the best results, with the training language coming in second for Indonesian and Thai, and fifth for Vietnamese. That being said, the Chinese test set performed comparatively well on almost all of the models, baseline and feature based, so this could simply be due to the fact that the posts in the Chinese data set were more explicitly emotional, or it could also be the case that the base XLM-R model is more suited for emotion classification in Chinese than other languages. The low-resource languages performed relatively poorly as expected, with most scores falling in the range of 38-46. Burmese and Tagalog performed best with the French model, achieving scores of 48.8 and 46.1 respectively. Khmer performed best on the English model with a score of 48.1.

When training on multiple Indo-European languages, the Indo-European test languages performed well, each with a F1 score of around 60, and both English and Portuguese performed better on this model than any other one. Burmese, which is not an Indo-European language, had a score of 50.4, its highest out of all the baseline and feature-based models. For the Sino-Tibetan model, Chinese performed well with a score of 62.1, though not as well as it did on the model solely trained on Chinese (66.5). For the Austronesian model, the Austronesian languages performed well, with Tagalog achieving its overall highest score of 58.0. Despite not being Austronesian languages, Burmese and Khmer also performed comparatively well on this model, and this is likely due to the fact that the languages this model was trained on had some of the highest amounts of data out of all the data sets. For the Austroasiatic model, both Vietnamese and Khmer performed relatively poorly compared to their performances on the other models. Each test language performed the best on the model that matched its language family except for the two Austroasiatic languages and Burmese.

Within the word order models, all but one language performed its best for its corresponding model. Additionally, the SVO model yielded the best results of any model for testing on Indonesian and Thai. English, French, and Portuguese still performed well, but the difference in performance between SVO and the next best model was smaller than between their results for the Indo-European model and any other language family model for each. The VSO model, which was trained solely on Tagalog, unsurprisingly performed the best on Tagalog data, but Vietnamese also got its best word order result with it, scoring a full 3 points higher than on its expected best, the SVO model.

For every testing language except one, the grapheme models produced the best performances for each language from the model with the matching feature value. In the Alphabetic model, Viet-

	IE	ST	AN	AA	TK
en	60.9	50.9	55.5	51.6	50.4
fr	59.2	51.1	54.0	52.3	52.3
pt	63.6	52.1	54.0	49.9	51.0
zh	56.6	62.1	59.4	57.0	58.2
id	49.2	46.6	51.7	49.5	46.4
vi	47.7	49.1	47.4	46.9	46.2
th	51.3	53.5	52.4	51.3	53.9
my	50.4	44.4	49.4	46.3	47.3
tl	49.6	38.1	58.0	43.2	38.8
km	48.0	47.5	49.3	43.0	47.2

Figure 2: Language Family Models
IE: Indo-European, ST: Sino-Tibetan
AN: Austronesian, AA: Austroasiatic, TK: Tai-Kadai

	SVO	SOV	VSO
en	59.9	57.1	51.4
fr	55.6	55.5	51.0
pt	57.8	55.9	53.1
zh	64.0	52.0	55.7
id	54.8	45.6	49.8
vi	46.2	44.2	49.2
th	55.0	51.4	46.6
my	46.7	47.5	44.7
tl	49.7	40.5	59.3
km	49.0	49.0	46.6

Figure 3: Word Order Models

	A	S	W
en	57.7	53.1	55.0
fr	59.5	54.3	53.0
pt	57.6	47.7	47.8
zh	58.4	58.6	66.5
id	51.6	48.2	49.9
vi	50.3	45.4	49.6
th	51.8	54.0	53.9
my	46.7	49.9	39.0
tl	48.8	43.5	38.6
km	50.1	47.5	40.1

Figure 4: Grapheme Models
A: Alphabetic, S: Syllabic, W: Word

namese and Khmer both had their best performance of any model, though Khmer is a syllabic language. Thai and Burmese had their second best performances on the syllabic model.

When testing each feature, languages primarily performed best on the model that corresponded with the feature value they had. Of course, there were a couple of exceptions for each feature, but three of the five model-language mismatches came from the languages defined as low-resource. This is almost certainly due to the lack of data, because if the full scope of a language isn't represented, it is less likely to perform well, even when a model is trained on a language or languages similar to it.

On the whole, no set of models for a feature consistently produced a significant margin of improvement for their corresponding languages to establish any feature as being explicitly indicative of emotion.

However, when applying the results to the creation of custom models for the low-resource languages, significant improvement can be seen. As shown in Table 2, the models for Burmese and Khmer showed significant improvement, besting their baseline models by 6.6 and 17.3 percent respectively. On the other hand, Tagalog performed significantly worse on the custom model. This makes sense, because despite being a low-resource language in terms of what XLM-R was pre-trained on, it actually had just as much data in the data set as a lot of the high-resource languages, whereas Burmese and Khmer each had less than 1000 examples in their data sets.

	Baseline	Custom
Burmese	43.3	49.9
Tagalog	59.3	47.0
Khmer	30.8	48.1

Table 2: Custom Models

6 Discussion and Conclusions

Overall, it can be concluded that XLM-R can be successfully used for emotion classification on low-resource languages. Even with no domain-specific training data, comparable performances can be achieved with models trained on specifically selected languages. However, there is still no replacement for having good domain-specific data, as was shown with the results for Tagalog's custom model.

In the future, this project could be improved my more varied data. In order to get results that conclusions can be more confidently drawn from, there needs to be no overlap in the languages that are trained and tested on. This would allow for good results to more likely be because of the contribution of language features rather than because a model was tested on the same language it was trained on.

It would also be interesting in future work to explore the boundaries of data scarcity, looking at how little data the general model can be pre-trained on for a language to still produce a respectable result when testing on strategically trained models like the ones discussed above.

If I were to do this project again, I would do more research into what features have been shown to be useful for the task of emotion classification so that the analysis could be more focused on what features work best for what languages rather than if a feature is helpful at all.

Another thing I would do is test the statistical significance of each result. This would give me a better idea of which results I should pay attention to, instead of looking at every time a model performed well compared to ones similar to it.

References

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Bangla text classification using transformers. *CoRR*, abs/2011.04446.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Omkar Gokhale, Shantanu Patankar, Onkar Litake, Aditya Mandke, and Dipali Kadam. 2022. Optimize$_{prime@dravidianlangtech-acl2022:Emotionanalysisintamil}$.

Sabit Hassan, Shaden Shaar, and Kareem Darwish. 2021. Cross-lingual emotion detection. *CoRR*, abs/2106.06017.

Sotiris Lamprinidis, Federico Bianchi, Daniel Hardt, and Dirk Hovy. 2021. Universal joy a dataset and results for classifying emotions across languages. volume 11th Workshop on Computational Approaches to Subjectivity, Sentiment Social Media Analysis (WASSA 2021). Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

D Vera, O Araque, and CA Iglesias. 2021. Gsi-upm at iberlef2021: Emotion analysis of spanish tweets by fine-tuning the xlm-roberta language model. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021). CEUR Workshop Proceedings, CEUR-WS, Málaga, Spain.*