MULTILINGUAL EMOJI PREDICTION

1 **Ashlesha, Bhagyasri, Divya, Sahana, Shweta, Spandana**
{ashu33, bkota, divsp, skowshik, shwetab, spatil33}@bu.edu

Github Link: https://github.com/dspoorthy/MultilingualEmojiPrediction/

1 Introduction

Emojis are a condensed part of communication that help understand and express intention and emotion. Emojis are widely used in social media and are effective at expressing emotions without the need for lengthy sentences. A major focus in the area of Natural Language Processing (NLP) is sentiment analysis[1][2][3][4], and several models pretrained on datasets like CC100, XLNI are available. The scope of this project is to fine-tune the multilingual pretrained models using the data obtained from Twitter to predict the corresponding emoji for each tweet. Multiple Transformer based models have been fine-tuned on the English and Spanish tweets and the resulting experiment outperformed the baseline of SemEval Competition [5] in terms of overall accuracy. The baseline scores from SemEval competition [6][7][8][9] are the results from a robust and high performing architecture using SVM and NN classifier. All the baseline F1 scores along with our predictions are tabulated below.

2 Approach

The scope of this project is testing discriminative and generative Transformer models on the same training and test dataset. Transformer-based architectures[10] are used in several areas of NLP. These architectures coupled with the emergence of large-scale pretrained models such as the latest BERT models have revolutionized the field of Natural Language Processing (NLP). Concepts such as transfer learning and pretrained language models have pushed the limits of language understanding and generation.[11]. Here we used four discriminative models (multi-BERT[12], Multilingual-MiniLM-L12-H384[13], XLM-Roberta[14] and mDeBERTa-V3 [15]) and three generative models (GPT-2[16], DistilGPT-2, GPT-NEO-125M[17]). Although discriminative models are preferred to generative ones for classification tasks, generative models give more details about the hidden parameters and can also output comparable results for certain data distributions

2.1 Datasets

Train Data: 500K tweets in English and 100K tweets in Spanish retrieved using Twitter API dated October 2015 to February 2017. The tweets themselves do not contain any emojis.
Test Data: 50k tweets generated in English and 10k tweets in Spanish are used to test the predictions.
Label Set: The label set comprises 20 and 19 most frequent emojis in English and Spanish respectively. This label set is different for English and Spanish data. A consolidated unique list of labels is generated and used for training to calculate the zero shot performance.

2.2 Analysis

As part of the preliminary data analysis, the word clouds are plotted for each of the English and Spanish training data. They give an idea about the word distribution and data skew in both the corpus. Preprocessing of the data that includes tokenization, removal of stop words, and cleaning up has been done to obtain a more meaningful understanding of the data. Refer to Figure 11 for a detailed understanding of the data.

2.3 Generative Classifier

As mentioned above we have experimented on three different generative models, GPT-2, GPT-NEO, and DistilGPT-2. More details about the model architecture and model specific details are mentioned in section 5. Here we outline the approach followed to fine tune each of these models. Generative models are not great at the task of classification and the way we give input for generative and discriminative models is quite different. Generative models are very good at identifying the underlying data distribution and then sampling from that particular distribution. While discriminative models are great at supervised tasks like classification where we have the pre-defined labels and our model can fit a function that can learn to predict the labels. In the general case of training or fine tuning, in a generative model like GPT-2, we give it raw text data without any labels and the model generates a word at a time in the form of predictions. If we are to perform a classification task using a Generative model we have to make use of the labels given and somehow make sure our model generates these labels given a sentence.
To achieve this, we converted our labels which are initially in the format of 0-19 for English and 0-18 for Spanish to tokens. For example for the emoji ❤️ the corresponding token is <_RED_HEART_>. After generating such tokens for both Spanish and English datasets, we add all these tokens at the end of the sentence of each of the lines of the training dataset of English

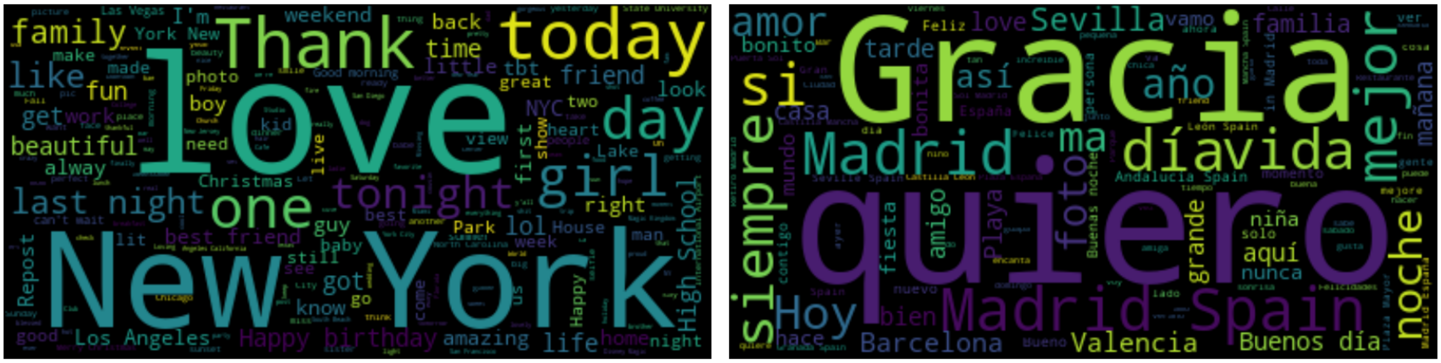

Figure 1: Fig on the left represents word cloud for English Dataset and Fig on the right represents word cloud for Spanish Dataset

and Spanish languages. Then we carry on the rest of the fine tuning process like a regular Generative Model fine tuning. We have used hugging face transformers GPT2Tokenizer for encoding each of these words in the text then used respective pre-trained models from hugging face and finetuned the models to the obtained encodings from the tokenizer. For each of these models, we have used AutoModelForSequenceClassification for getting the respective classified labels. We have also used AutoModelForSequenceLanguageModeling where during test time we generate a few tokens and then we select the first token that was generated from among the sequence of other words since a model will not just output tokens, it could also output other words from the distribution of the training dataset. We use this as our label and calculate the accuracies using ground truth.

2.4 Discriminative Classifier

Discriminative models chosen for the task are MultiBERT, XLMRoberta, Multilingual-Mini-LM, and mDeBERTa-V3. This clearly shows the evolution of BERT based models through 2018 - to date. Our experiment showcases the results of these various models and how the accuracy has been improved. All the multilingual models chosen for the task are uncased since we are working with tweets and our sole objective is emoji prediction. Although discriminative models are more efficient when compared to generative for classification tasks, the trade-off here is the runtime. Discriminative models such as Roberta and mDeberta are pretrained on huge datasets of 100+ languages and have around 86M backbone parameters. These are improvements over the bert model which is proved to be undertrained. XLM-Roberta is a multilingual model which is pretrained on raw text and is a self-supervised model with an automatic process to generate inputs and labels from the input text. On the hand, mDerberta uses disentangled attention and enhanced decoder which typically used two vectors(content and relative position) per token rather than the typical single vector used by most Bert models. Although the runtime is significantly more, the Deberta model surpassed the hum prediction baseline and is one of the best models for classification as of today.

The input raw texts are processed using the model tokenizers which in the case of multiBert uses 110k shared wordpiece vocabularies and XLM-Roberta uses a pretrained base tokenizer based on sentence piece. the mDeberta-V3 model uses a new sentencepiece-based tokenizer that uses new vocabulary built based on the training data which increases the efficiency. These tokenizers can be initialized using the AutoTokenizer from pretrained by mentioning the checkpoint or from their respective model classes. Fine tuning the pretrained discriminative models is a straightforward approach wherein we load the model from the AutoModelForSequenceClassification to train on the dataset. We used flat accuracy which is a similar measure to Sklearn's accuracy package to calculate the model accuracies. Due to the computational limitations, the models were run for a max of 1/2 epochs and then are used to make predictions on the test data.

3 Evaluation metrics

- F1 scores for each emoji:

$$f_1 = 2 * \frac{precision * recall}{precision + recall}$$

- Top 1 accuracy for English and Spanish: Top 1 accuracy is the conventional accuracy. The prediction with the highest probability is used to calculate the accuracy.

$$Accuracy = \frac{correct\ predictions}{total\ predictions}$$

- Top 3 accuracy for English and Spanish: Top 3 accuracy uses the top 3 predictions of the model. If the true label is present in the top 3 predictions, the model is assumed to be predicting the correct label.
- Top 1 and top 3 zero shot accuracy for Spanish on English and English on Spanish: The model trained on English train data is tested on the Spanish test set and vice–versa. Top k accuracy is performed on the resulting predictions.

4 Experiments and results

To compare various performances of all the selected models we have tabulated all the results we obtained in Tables 1-4. All the models we used, a little description, and the intuition behind why we used those models is described below in section 4.1.

4.1 Model selection

4.1.1 Generative models

- **GPT-2**: This model, introduced by OpenAI, is a transformer based model that uses attention based models in place of convolutional based architecture. This model has about 1.5 billion parameters and is very good at the task of text generation. The reasoning behind using this model is to test the efficacy of this generative model and try to see how it performs on classification tasks since this is one of the state of the art generative models existing in NLP literature.
- **DistilGPT-2**: DistilGPT-2 weighs 37% less, and is twice as fast as its OpenAI counterpart while keeping the same generative power. One of the primary reasons why we chose this model is to compare its results to see if we can achieve the same performance using less compute power and reduced training time.
- **GPT-Neo-125M**: GPT-NEO is a mesh TensorFlow library that is an implementation of data-parallel models like GPT-3. Since GPT-3 does not have open access we wanted to try a model that is close to GPT-3's performance to see how it would perform on our data. We chose the 125M model because of constraints in computing power and training time.

4.1.2 Discriminative models

- **MultiBERT**: This is a multilingual bert based model introduced in 2018 and the first bidirectional model. The model is pretrained on the top 102 languages using Wikipedia text using Masked Language Modeling(MLM). As a result, languages with fewer resources on Wikipedia are undertrained which is a limitation.
- **Multilingual-MiniLM-L12-H384**: This is a light model that was introduced in 2020 as an alternative to heavy pre-trained models like Bert and XLM-Roberta which contain hundreds of millions of parameters. This model has around 21M parameters. Multilingual MiniLM uses the same tokenizer as XLM-R. But the Transformer architecture of this model is the same as BERT. The main reason to choose this model is to compare its performance with its heavy counterparts like XLM Roberta.
- **XLM-Roberta**: XLM-Roberta is a Roberta based model and is an enhancement over the bert models. The model is pretrained on 2.5TB of filtered common crawl data containing 100 languages. The data are pure raw text with no human labeling. The hidden layer of the model has 768 dimensions with 12 layers of transformers. The advantage of this model is that it takes less time to train as compared to the bert model as it is robustly optimized. The model randomly masks 15% of the words and the model uses an attention mechanism to predict the masked words which increase the efficiency of the model.
- **mDeBERTa V3**: Currently Deberta is the only model to have surpassed the human predictions accuracy. This model is an improvement over mDeberta which uses ELECTRA style training and gradient disentangled embedding sharing which improved the model performance significantly over the Roberta models by a minimum of 2% on 80GB training data. The model comes with 86M backbone parameters with a vocabulary size of 250K. It also introduces 190M parameters in the Embedding layer. This model was pretrained using 2.5T CC100 data similar to XLM-R.

4.2 Discussion and Analysis of Performances

- By looking at consolidated results from Table 1 4.2 we see that our model mDBERTA-V3 outperforms all the other models (both generative and discriminative) including baseline by a margin of 3%. Other models like XLM-Roberta and miniLM, even though they did not perform the best, still outperformed baseline by a small margin. Extensive and better hyper-parameter tuning and an increase in training time could result in much better performances of these models.
- We see that Generative models could not outperform baseline and did much worse in comparison to the discriminative models. This was expected. Generative models are pretty good at capturing the underlying distribution of the data but when it comes to solving a problem of classification, due to the assumption made by the generative models, their performance is lower than that of discriminative models.
- From table 2 2, we can infer that most of our models outperformed baseline in almost all emoji categories. There were few emojis for which baseline accuracy was higher but that could be due to the difference in the distribution of the training and test data sets. Another possibility might be the similarity in the emojis which might cause a wrong prediction but in reality, can be a good one.
- From table 3 3, Our models did not perform that well on Spanish data but they were very close to the baseline. The training data available for the Spanish dataset is relatively low when compared to the English data which might be insufficient for the model fine tuning. GPT models performed poorly again in comparison to BERT based models on the Spanish dataset, which could be attributed to the fact that GPT models are mainly trained on an English corpus while the BERT models we chose were all multilingual models and were trained on a corpus of languages including Spanish and English.
- From table 4 4.2, we present the results from the zero-shot accuracy calculated by testing the model trained on the English dataset on the Spanish test set and vice versa. Although the accuracy is not as high as in the other scenario when we consider the top 3 instead of just the top 1, it shows a significantly higher value. More training data might also help improve the zero-shot accuracy.

Model	**Top1 Acc English**	**Top3 Acc English**	**Top1 Acc Spanish**	**Top3 Acc Spanish**
Baseline	47.09	-	37.27	-
GPT2	43	63.6	34.16	55.8
DistilGPT2	42	62.2	33.4	54.8
GPT Neo	29	48.5	26	44.1
multiBERT	39.02	63.46	27.69	48.18
XLM Roberta	49.58	70.8	40.62	60.43
mDeBERTa	**50.28**	**71.32**	**41.41**	**61.25**
MiniLM	47.7	68.2	38.6	57.5

Table 1: **Consolidated results**: *Here we have the average test accuracies for each of the models we performed experiments on (both generative and discriminative). The first column corresponds to the model used, the second column corresponds to average test accuracies for all the emojis only when one label is considered for the English dataset, the third column corresponds to average test accuracies for all the emojis when the first three labels are considered for the English dataset, the fourth column corresponds to average test accuracies for all the emojis only when one label is considered for the Spanish dataset, the fifth column corresponds to average test accuracies for all the emojis when the first three labels are considered for the Spanish dataset.*

Table 2: F1 SCORES PER EMOJI FOR ENGLISH DATASET

Model	**Baseline**	**multi BERT**	**XLM Roberta**	**mDeBERTa**	**MiniLM**	**GPT2**	**DistilGPT2**	**GPT Neo**
❤	87.8	72	88	**88**	87	69	68	46
😍	37.8	28	40	**41**	38	36	35	22
😂	47.1	52	50	**52**	48	44	42	37
💕	26.9	0	28	**28**	28	24	22	0
🔥	55.5	57	59	**59**	56	50	49	34
😊	16.2	5	18	**19**	17	14	13	1
😎	22.6	14	**23**	21	21	20	19	4
✨	36.2	21	37	**37**	35	30	30	4
💙	**24**	5	18	12	13	16	18	0
😘	22.2	1	**23**	22	19	17	17	0
📸	38.4	8	43	**45**	39	22	22	0
🇺🇸	64.7	55	67	**69**	66	59	58	28
☀	63.7	52	**71**	70	69	59	58	27
💜	**17.1**	3	3	0	6	6	9	0
😉	**13**	2	11	4	11	11	11	0
💯	29.2	17	30	**34**	28	27	27	0
🎄	73.6	**79**	74	74	73	65	66	39
📷	40	**61**	17	9	4	33	31	4
😜	**9**	0	0	0	0	3	5	0
😁	**14.3**	0	0	4	1	7	6	0

Table 3: F1 SCORES PER EMOJI FOR SPANISH DATASET

Model	Baseline	multi BERT	XLM Roberta	mDeBERTa	MiniLM	GPT2	DistilGPT2	GPT Neo
🖤	**69.6**	36	68	68	66	56	56	41
😍	37.3	32	38	**39**	36	31	32	22
😂	53.4	48	56	**58**	54	48	46	31
💕	8.5	1	1	**17**	0	3	9	4
😊	14.9	3	**17**	13	12	9	11	9
😎	**14.7**	0	0	0	0	2	6	1
✨	**20**	5	8	2	10	7	12	4
💙	**14.2**	2	0	0	0	0	1	0
😘	**26.9**	2	25	25	19	19	21	10
👌	**13**	0	1	2	4	9	6	2
🇪🇸	49.9	46	51	48	**50**	43	37	27
💪	**39.8**	32	37	34	38	29	28	15
💜	**6.8**	2	0	0	0	0	0	0
😉	**16.3**	1	5	15	3	2	3	7
💘	**8.6**	0	0	0	0	0	1	0
💞	**5.6**	0	0	0	0	0	0	0
🎶	**23.7**	15	22	17	20	14	18	11
😜	**7.7**	0	0	0	0	1	1	1
😁	**5.1**	0	0	0	0	0	0	2

Model	Top1 Acc on Spanish English	Top3 Acc on Spanish English	Top1 Acc on English Spanish	Top3 Acc on English Spanish
GPT2	28	47.4	26	39.4
DistilGPT2	27	46.2	24	36.8
multiBERT	25.45	44.91	22.82	36.81
XLM Roberta	31.01	42.77	34.94	52.69
mDeBERTa	**32.41**	42.93	**35.48**	**53.31**
MiniLM	32.22	**48.51**	29.1	40.2

Table 4: **Zero–shot results** *Here we have the zero test accuracies for each of the models we performed experiments on (both generative and discriminative). The first column corresponds to the model used, the second column corresponds to average test accuracies for all the emojis only when one label is considered for Zero shot performance on English dataset trained on Spanish Dataset, the third column corresponds to average test accuracies for all the emojis when first three labels are considered for Zero shot performance on English dataset trained on Spanish Dataset, the fourth column corresponds to average test accuracies for all the emojis only when one label is considered for Zero shot performance on Spanish dataset trained on English Dataset, the fifth column corresponds to average test accuracies for all the emojis when first three labels are considered for Zero shot performance on Spanish dataset trained on English Dataset.*

5 Conclusion

- From the results, we can observe that Discriminative models outperform Generative models when it comes to text classification.
- There is also a significant increase in the accuracy when the top 3 emoji predictions are weighed against the ground truth.

References

[1] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[2] Tan Thongtan and Tanasanee Phienthrakul. Sentiment classification using document embeddings trained with cosine similarity. In *ACL*, 2019.

[3] Zijun Sun, Chun Fan, Qinghong Han, Xiaofei Sun, Yuxian Meng, Fei Wu, and Jiwei Li. Self-explaining structures improve nlp models, 2020.

[4] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[5] Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. SemEval 2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 24–33, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[6] Çağrı Çöltekin and Taraka Rama. Tübingen-Oslo at SemEval-2018 task 2: SVMs perform better than RNNs at emoji prediction. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 34—38, New Orleans, LA, United States, 2018. Association for Computational Linguistics.

[7] Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. 04 2018.

[8] Man Liu. Emonlp at iest 2018: An ensemble of deep learning models and gradient boosting regression tree for implicit emotion prediction in tweets. pages 201–204, 01 2018.

[9] Jonathan Beaulieu and Dennis Asamoah Owusu. UMDuluth-CS8761 at SemEval-2018 task 2: Emojis: Too many choices? In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 400–404, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[11] Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. Transformers: ”the end of history” for nlp? 2021.

[12] Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. The multiberts: Bert reproductions for robustness analysis, 2021.

[13] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.

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

[15] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.

[16] Vanya Cohen and Aaron Gokaslan. Opengpt-2: Open language models and implications of generated text. *XRDS*, 27(1):26–30, sep 2020.

[17] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.