AngriBERT - Multi-Label Emotion Classifier

Devesh Khandelwal
University of California - Berkeley
deveshkhandelwal@berkeley.edu

Manohar Madhira
University of California - Berkeley
rmadhira@berkeley.edu

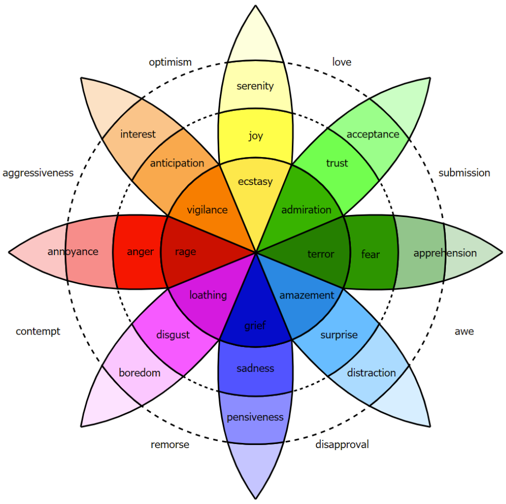
Abstract

Understanding emotions expressed in a text has a wide range of applications. In a business context, understanding users' emotions in online forums, support tickets, and social media platforms could help companies improve their focus on things that matter most. However, understanding emotions from text is a challenging problem. Creating a large domain-specific dataset takes time and is constrained by resources(schedule, budget and compute). In this paper, we evaluate if transfer learning across domains could reduce the need for domain-specific data. We train a range of models on disparate datasets and then evaluate their performance on an unrelated target dataset. We also demonstrate that a pre-trained model from another domain not only reduces the number of samples needed to train a model to achieve a performance benchmark but also outperforms(comparable) model(s) which were trained and fine-tuned with domain-specific data only. The performance behavior though is dependent on the choice of the pre-trained data.

1 Introduction

Emotion expression is central to human interaction. Emotions expressed in communications have large-scale implications for automated systems and machine learning models. Understanding the emotion expressed in a text could help organizations make better decisions. e.g. support tickets are likely to have negative sentiment, but understanding whether a customer is frustrated, or angry can help businesses route the ticket more appropriately.

There are many emotion classification models. The most popular ones are: Ekman's (Ekman, 1992) six basic emotions(1 positive and five negative). Plutchik's (Plutchik, 1980) wheel of emotions: eight primary emotions with three intensity levels and eight relations, providing 32 emotions in total. Common datasets in the field use multiple variations of these models. (Bostan and Klinger, 2018) proposes a unified model for classifying emotions, standardizing their mapping on different corpora to Plutchik's model wherever possible. Figure 1 shows Plutchik's wheel of emotions.



Source: https://commons.wikimedia.org/w/index.php?curid=13285286/

Figure 1: Plutchik's wheel of emotions

According to Rosalind Picard (Picard, 1997), to make computers genuinely intelligent and to interact naturally with us, we must give them the ability to recognize, understand, even to have and express emotions.

To teach computers to recognize and express emotions, we need a lot of labeled data. However, despite the need, most annotated datasets are small and sometimes non-existent in a domain of interest. This paper shows that models trained on corpora with similar distribution perform better, train faster, and reduce cost(compute and annotation)by needing smaller datasets in the target domain. This transfer of knowledge, however, does not apply universally. We demonstrate that when the corpora don't follow a similar distribution (like tweets and dialogues) the performance of a model degrades.

Our contribution in this paper is two-fold: leverage transfer learning approaches to evaluate performance across domains and provide a framework for assessing and creating cross-domain pre-trained models.

2 Background

In the past two decades researchers have published several datasets for emotion classification.'Fairy-Tales' (Alm et al., 2005) a corpus of 14771 sentences from fairy tales,'Headlines' (Strapparava and Mihalcea, 2007) a corpus of 2462 headlines. 'Electoral Tweets' Saif M Mohammad and Martin (2015) - 4185 tweets, Crowdflower (2016) [1] - 39740 tweets, Schuff et al. (2017) - 9107 tweets, ŚEMEVAL' citetSEMEVAL2018Task1 - 10983 tweets are some examples.

However, these datasets 1) follow different annotation schema. 2) have different file formats and 3) belong to different domains. (Bostan and Klinger, 2018) created a unified emotion classification model by standardizing the annotations to the (Plutchik, 1980) model wherever possible. We build on the work done by (Demszky et al., 2020) and (Bostan and Klinger, 2018). (Demszky et al., 2020) provides a ”unified dataset” and tests the performance of its annotated Reddit comment dataset against individual datasets published by the (Bostan and Klinger, 2018). However, it doesn't explore the effect of unifying the data and evaluate its predictive power. (Bostan and Klinger, 2018) publishes the performance results on maximum entropy classifiers as implemented in scikit-learn with bag-of-words (BOW) features for these experiments. We extend these approaches by:

1. Consolidating diverse datasets from **GoEmotions** (Demszky et al., 2020) containing Reddit comments, **SEMEVAL** (Mohammad et al., 2018), containing tweets and **Friends** (Chen et al., 2018) containing dialog into different corpora using Ekman's model of 6 basic emotions + noemotion.

2. Testing if transformer-based models pre-trained for emotions from these datasets can perform well in a new domain that uses structured language like **Tales** (Alm et al., 2005) containing lines from fairy tales?

3 Data Selection

3.1 Dataset Overview

Go Emotions (GO)

GoEmotions[2] (**GO**) dataset contains Reddit comments annotated with 27 emotions + ”no emotion”/neutral.This dataset contains 43410 train and 5426 dev records.

Friends Emotion Lines (FR)

Friends[3] contains emotions annotated on dialog from Friends TV Series using Ekman's basic emotions.The dataset contains 10561 train and 1178 dev records.

SEMEVAL 2018 (SEM)

SEMEVAL 2018[4] contains tweets annotated on 11 categories. Dataset contains 6838 train and 886 dev records.

Fairy Tales (TALES)

Tales[5] contains sentences from fairy tales books annotated using Ekman's model. This dataset doesn't come with predefined splits(train, dev, and test). We manually create 80:10:10 random splits of the total data. The resulting dataset contains 11817 train, and 1477 dev and test records each.

3.2 Data Preparation

We map the annotations on the GO and SEM datasets to Ekman's model. Table 1 shows the mapping definitions from GO (specified in (Demszky et al., 2020)), SEM (specified in (Bostan and Klinger, 2018). FR and TALES are already annotated using Ekman's model and hence need not be re-mapped.

Figure 2 shows the distribution of emotions after consolidation. Our datasets are highly imbalanced. We retain this imbalance in classes as it is more representative of real-world scenarios. We evaluate and report on F1-micro, which is a recommended metric for imbalanced datasets.

[1] https://data.world/crowdflower/sentiment-analysis-in-text

[2] https://github.com/google-research/google-research/tree/master/goemotions/data

[3] http://doraemon.iis.sinica.edu.tw/emotionlines/download.html

[4] https://competitions.codalab.org/competitions/17751

[5] https://github.com/sarnthil/unify-emotion-datasets

Emotion	Dataset	Consolidation
joy	GO	All positive emotions: admiration, amusement, approval, caring, desire, excitement, gratitude, joy, love, optimism, pride, relief
	SEM	anticipation disgust fear joy love optimism
anger	GO	anger, annoyance, disapproval
	SEM	anger
disgust	GO	disgust
	SEM	disgust
fear	GO	fear nervousness
	SEM	fear
sadness	GO	sadness, disappointment, embarrassment, grief, remorse
	SEM	sadness, pessimism
surprise	GO	surprise, confusion, curiosity, realization
	SEM	surprise
noemo	GO	noemo
	SEM	Not Applicable

Table 1: Emotion Mapping Between Datasets

All the datasets are multi-labeled except TALES. Despite that, we use TALES as our target domain dataset to check whether transfer learning techniques are effective in this scenario.

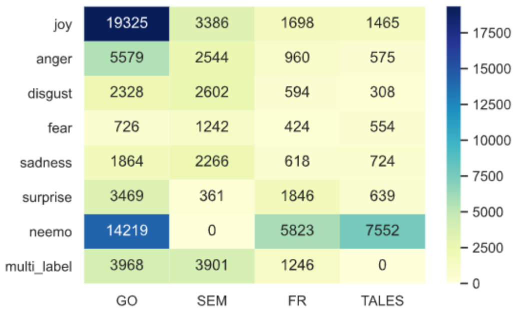


Figure 2: Count distribution across emotions consolidated to Ekman's basic model

4 Experiments

4.1 Our Approach

To study the effect of transfer learning from training on a different corpus, we train and fine-tune our models using HuggingFace's DistilBERT[6] (Sanh et al., 2019). Using DistilBERT simplifies the training process compared to $BERT_{base}$ and $BERT_{large}$ and also simulates a constrained environment. We follow Devlin et al. (2018) guidelines to use the embedding corresponding to the input [CLS] token for classification purposes. To generate a multi-label classification, we pass the final layer [CLS] embedding through a fully connected layer with a sigmoid activation for classification. We also preserve the case of our input sentences by using a case sensitive tokenizer- **'distilbert-base-cased'**

We train and tune hyperparameters on the largest corpus, the *GO* dataset and then keep hyperparameters constant on subsequent models to compare their performance.

To simulate the scenario where the resources(annotated data and compute) are scarce-we train, multiple independent models by incrementally adding data from the *TALES* corpus and comparing their performance to a *TARGET* model-one which is exclusively trained on *TALES* data with the same number of samples. The goal is to compare the performance of models trained with transfer-learning techniques to the *TARGET* model with the same sample size. To reduce the training time and compute cost we also perform various experiments by freezing different layers of our models.

4.2 Hyperparameters

To arrive at the best set of hyperparameters we begin by finetuning *GO*. We take the starting hyperparameter values from Demszky et al. (2020) and further finetune to optimize results. We primarily focus on adjusting the learning rate, batch size, and epochs for our scenario. We train the *GO* dataset with learning rates from [1e-5 to 5e-5] in increments of 1e-5, epochs [3,4,5], and batch sizes of [2,4,8,16,32]. We find that the learning rate of 4e-5 with 4 epochs gives the best performance. We favor a batch size of 16 over 8 as it generalizes better on our validation dataset. We retain the dropout rate of 0.2 in the pre-classifier layer and keep the maximum token length to 512 to account for large sentences in the corpora. Demszky et al. (2020).

[6] https://huggingface.co/transformers/model_doc/distilbert.html

4.3 Experiments

4.3.1 Target Creation

As discussed in an earlier section, we first train multiple independent models with different sample sizes - [100, 500, 1000, 1500, 2000, 3000, 5000, 8000, and ALL] on the *TALES* to establish the *TARGET*. We don't save the models from the previous runs and train from scratch each time a new sample size is selected. This approach allows us to arrive at a 'benchmark' for a sample size needed to achieve the desired level of metric, which is context-dependent.

4.3.2 End to End Training

To conduct Transfer Learning experiments, we train multiple models on the following datasets.

model1 *GO* – trained on GoEmotions dataset

model2 *GO_FR* –trained on GoEmotions and Friends combined dataset.

model3 *GO_SEM* –trained on GoEmotions and SEMEVAL 2018 dataset.

model4 *GO_SEM_FR* – trained on GoEmotions, Friends, and SEMEVAL 2018 datasets

We then take each one of these trained models(model1,model2,model3,model4) and re-train them by incrementally adding the same number of samples from TALES that were initially used in the TARGET creation. This helps us to compare the performance of these models against TARGET when trained on identical sample sizes from TALES. This is the main idea behind testing our hypothesis - if we can use pre-trained models from an unrelated domain and achieve satisfactory results in our domain of choice with fewer training samples.

4.3.3 Additional Architectural Choices -Freezing Layers

We also wanted to see whether we can save computing costs by shortening the training time. To study this effect we randomly freeze different layers in our DistilBERT architecture and conduct experiments. Since freezing of layers results in fewer weight adjustments during back-propagation, our idea is to conduct experiments with additional architectural choices. We conduct 16 experiments and our top 3 model architecture choices are listed below: model4 *GEMB* – GoEmotions with embedding(the encoder output) layers frozen

model5 *GEMB_FR_3_5* – GoEmotions + Friends with embedding(the encoder output)and layers 3 to 5 frozen

model6 *GEMB_SEMFR_0_5* – GoEmotions, SEMEVAL 2018, Friends with all layers frozen. In this set up the subsequent training will only adjust weights in the final fully connected and classification layers. We then use a similar approach mentioned earlier in the previous sub section to compare the performance of these models(model4,model5,model6) against the TARGET.

5 Results and Discussion

Table 2 compares various models.To demonstrate the performance boost obtained with neural models, we also include the score obtained on a maximum entropy classifier(MAXENT)- the primary classification model in (Bostan and Klinger, 2018).

We note that our neural models could match the performance of the MAXENT classifier with 5 percent of the data. We also note that when the target domain data is small(500) our models outperform the TARGET. This behavior changes a little as we add more domain-specific data but as we approach the maximum sample size on TALES, the majority of our models outperform the TARGET.

Model	ALL[a]	D_ALL[b]	500[c]	D_500[b]
MaxEnt	0.5400			
TALES	0.8960[e]		0.5389	
GO_FR	0.9139	**0.0179**	0.5842	**0.0453**
GO	0.9142	**0.0182**	0.5779	0.0390
GEMB_FR_3_5	0.9019	0.0059	0.5652	0.0263
GO_SEM_FR	0.9084	0.0124	0.6001	**0.0612**
GO_SEM	0.9052	0.0092	0.5845	**0.0456**
GEMB	0.9062	0.0102	0.5565	0.0176
GEMB_SEMFR_0_5	0.5628	-0.3332	0.5451	0.0062

Table 2: F1-Minor Scores Across Models Evaluated on TALES

Please use the key below for the table above:

- [a] score when trained entire TALES set.
- [b] difference in score between TARGET and model when trained on entire TALES set.
- [c] score when trained on 500 samples from TALES.
- [d] difference in score between TARGET and model when trained on 500
- [e] TARGET score when trained entire TALES set.

As started earlier we train with 3 epochs and calculate the metric once on our target-domain data(TALES) as against calculating the metric multiple times and reporting the mean/median.

The sections below elaborates more on our results and approach to hypothesis testing - combining disparate datasets, training a model on this combined data, and then using it to create a series of models by incrementally adding domain-specific data from TALES. This helps us to benchmark the performance of our models against the TARGET.

5.1 GO_FR

We create a model on combined FR + GO datasets and then create a series of models by adding samples from TALES.As evident from 3 and table 5 these model outperforms the TARGET when trained on the same sample size from TALES . We notice that the TARGET's performance is marginally higher until we add 1500 samples from TALES to training but past this sample size, our models consistently outperform the TARGET. We also observe that both these models and TARGET have almost identical scores when trained on 2000 -5000 training samples from TALES but as we add more training samples (GO_FR) based models produce better results. It is difficult to determine the cause of this behavior, but we suspect our architecture (DistilBERT)needs at least 1500 samples from the domain corpus to pick up signals. This pattern is consistent among all models(also see the conclusion section).

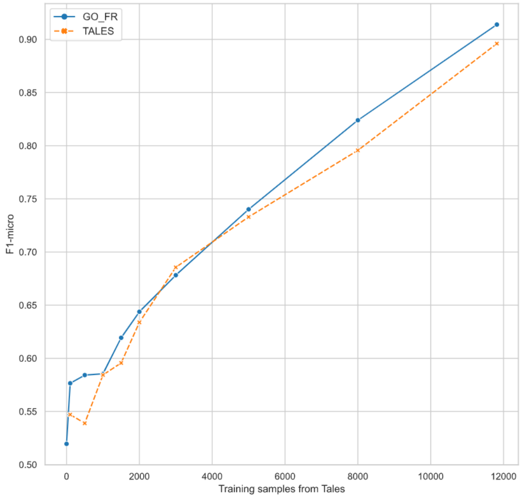


Figure 3: F1-Micro of GO_FR and TARGET on different sample sizes of TALES dataset

5.2 GO

As shown in Figure 4, we find that when we use a model trained on GO as a base and then create a series of models on top by adding TALES data it outperforms the TARGET. These models are adversely affected when we train on sample sizes in the range of 1500-3000 from TALES

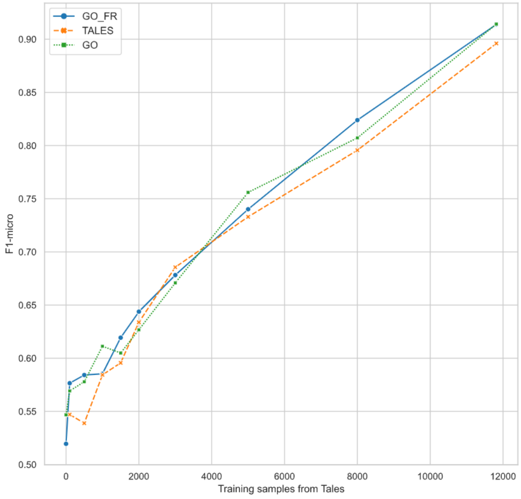


Figure 4: F1-Micro of GO,GO_FR,TALES on different samples from TALES

5.3 GO_SEM_FR

We also see from Figure 5 that models trained on GO as base outperform the models that were trained by adding TALES on top of a model trained on the GO_SEM_FR combined. These models GO_SEM_FR category perform better at a very low sample size, but we notice that they suffer immensely as we add more TALES data and re-train the models.

These results are interesting but a little counter-intuitive. We generally expect that additional training data would lead to a better performing model. We observe that combining corpora yields better performance on the category of models that are trained on small samples of TALES data. But, the choice of training corpus impacts the model performance. In the next section, we cover a few provide a few guidelines one could consider when selecting a training corpus to use for transfer learning. Please see Appendix A for additional results.

5.4 Analysis

5.4.1 Word Distribution in Training Samples

Log Odds ratio across different corpora show that SEMEVAL corpus differs significantly from

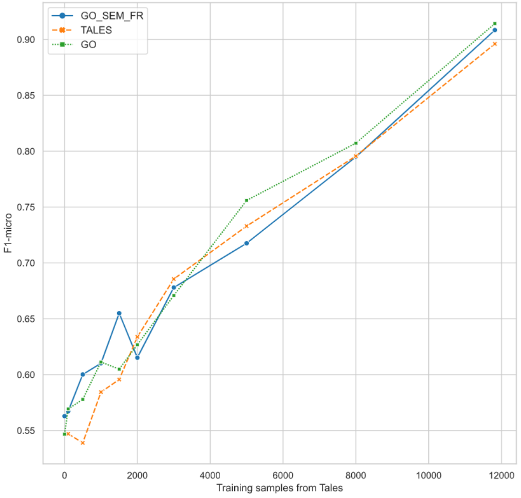


Figure 5: F1-Micro of GO,GO_FR,GO_SEM_FR on different samples from TALES

the rest. It has words like (thenicebot,bodyfit67, chr1smclaughlin,behappy, goodday). This prompts the need for further analysis. Table 3 shows a sample of the top 5 words associated with the emotion Fear. We tried to use regular expressions to pre-process SEMEVAL and the corresponding results are shown in the column SEM_CLEAN. The words in SEM_CLEAN could be user handles. Our analysis indicates that in general Twitter data should be pre-processed before applying to any NLP task that crosses a domain.Please see Table 6 in appendix for complete list of Log-Odds results.

	SEM	SEM_CLEAN	TALES
Fear	bodyfit67	animator	terrified
	senses	cparksnum	uneasy
	seanunfiltered	vovimprgel	horror
	frighten	seanunfiltered	frightened
	ahsroanoke	spookytv	atwitter

Table 3: Sample of top 5 words using Log-Odds

5.4.2 Perplexity Between Corpora

We compare the similarities between the corpora by calculating perplexity values on a language model. We train a tri-gram language model on the TALES data and then use this model to generate perplexity scores on randomly selected sentences from the other corpora.Table 4 gives the perplexity values when we randomly select 5 sentences from other corpora.

Our results show that SEMEVAL's perplexity is higher than TALES'. This difference in perplexity scores explains why we see a performance drop when we train models with SEMEVAL and use them as a base to perform transfer-learning experiments.

Corpus	Perplexity
FR	3938.60
GO	4917.31
SEM	6473.44

Table 4: Perplexity values on a tri-gram language model trained with Tales(k=.01)

6 Conclusion

We show that when the data is scarce in a target domain, it is possible to achieve better results by training an emotion analysis model in another domain and then use transfer learning techniques to train models by adding samples from the target domain. We have also shown that in constrained situations, training time can be reduced by freezing intermediate layers.

Our results show the potential and pave the path for future work. Future work could use this approach and explore the performance on larger models like BERT$_{large}$, ROBERTa or XLnet.

Future work could also include an analysis of the models' performance and could further investigate the reasons behind the better performance of TARGET when the sample-size is in the range 4k-6k .

To enable such work, we are releasing our code to public domain[7].

Acknowledgments

We thank Prof. Mark Butler for all his support and guidance for this project.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics.

Laura-Ana-Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th Inter-*

[7] https://github.com/rmadhira86/w266-finalproject/

national Conference on Computational Linguistics, pages 2104–2119. Association for Computational Linguistics.

Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Ting-Hao, Huang, and Lun-Wei Ku. 2018. Emotionlines: An emotion corpus of multi-party conversations.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions.

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*.

Paul Ekman. 1992. An arugment for basic emotions. *Cognition Emotion*.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Rosaline Picard. 1997. *Affective Computing*. The MIT Press, Cambridge, MA,USA.

Robert Plutchik. 1980. *Theories of emotion*. Elsevier.

Svetlana Kiritchenko Saif M Mohammad, Xiaodan Zhu and Joel Martin. 2015. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing Management*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Hendrik Schuff, Jeremy Barnes, Julian Mohme, Sebastian Pad'o, and Roman Klinger. 2017. Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 13–23.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.

Appendices

A Performance of All Models: Graphs

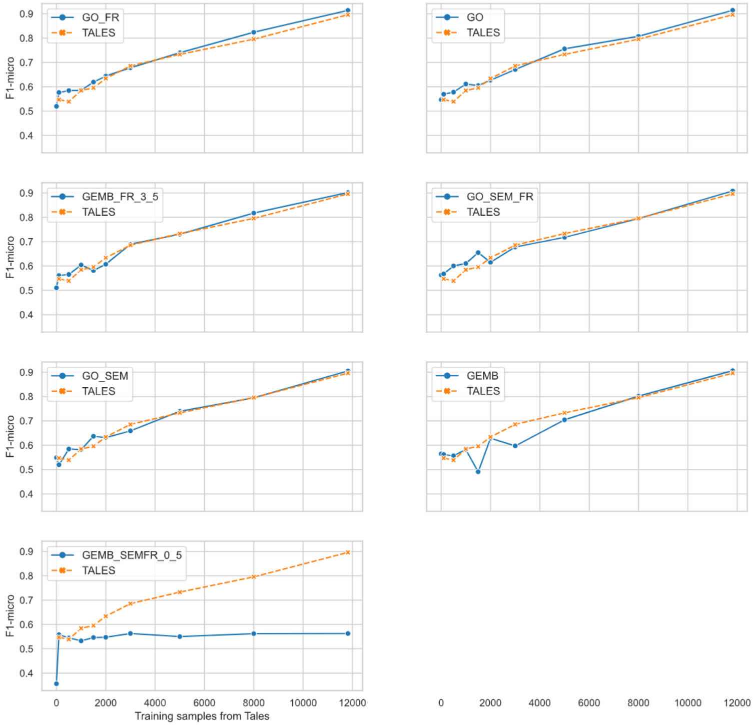


Figure 6: Performance of various models and TARGET on different sample sizes of TALES dataset

B A comparison of few Base Models when trained on different sample sizes on TALES

name	TALES	GO_FR	D_GO_FR
train_count			
0		0.5195	
100	0.5471	0.5765	0.0295
500	0.5389	0.5842	0.0453
1000	0.5844	0.5854	0.0010
1500	0.5956	0.6193	0.0237
2000	0.6338	0.6438	0.0100
3000	0.6856	0.6782	-0.0074
5000	0.7330	0.7401	0.0072
8000	0.7956	0.8239	0.0283
11817	0.8960	0.9139	0.0179

Table 5: F1 Micro of TARGET and other models on various sample sizes

Please refer to section 5 for an explanation of model configuration.

C Top 5 words by emotion

emotion	TALES	GO	SEM	FR
joy	happily:68.5	congratulations:29.6	thenicebot:29.5	yay:58.1
	kywitt:68.5	excellent:28.8	behappy:21.5	incredible:40.7
	rejoiced:68.5	thank:23.6	lively:19.8	woo:34.8
	delighted:59.2	funnier:22.2	goodday:18.8	soo:34.8
	glad:58.1	awesome:20.4	gift:18.8	dum:34.8
anger	rage:155.2	shitshow:28.4	azerbaijan:11.3	punk:33.6
	furious:97.0	pittsburgh:28.4	overpowered:7.5	wesley:33.6
	cuck:97.0	unh:28.4	biggkatz:7.5	angry:33.6
	annoyed:77.6	ding:28.4	rodrigo:7.5	charge:33.6
	fury:77.6	revisionist:28.4	stub:7.5	suggestion:33.6
disgust	pooh:174.6	embarassed:70.9	tasted:11.0	ew:181.2
	disgusted:131.0	disgusting:66.4	cherrivarisco:7.3	eww:60.4
	teased:131.0	embarrassing:57.6	nasboat:7.3	boobs:60.4
	expensive:131.0	embarassing:53.2	dada:7.3	violated:45.3
	omitted:87.3	troo:53.2	custom:7.3	moms:45.3
fear	terrified:209.8	terrified:208.6	bodyfit67:27.5	wa:72.5
	uneasy:152.6	petrified:187.7	senses:27.5	vulnerability:48.4
	horror:95.3	horrified:187.7	seanunfiltered:27.5	terrified:48.4
	frightened:88.2	terrors:187.7	frighten:27.5	hangs:48.4
	atwitter:57.2	neigh:187.7	ahsroanoke:27.5	weights:48.4
sadness	sorrowfully:109.9	saddest:86.8	gray:13.9	cramp:59.0
	mournfully:109.9	madridista:71.0	kpop:13.9	28:44.2
	wept:74.6	3ds:71.0	constitutional:13.9	hopes:44.2
	mourned:68.7	rosetta:71.0	georges:13.9	knowing:44.2
	pitied:55.0	bearer:71.0	groovydadad:13.9	ye:44.2
surprise	shocked:89.6	import:48.0	mania:68.8	whoa:37.4
	ow:71.7	uncanny:36.0	arrivals:68.8	oww:27.0
	astonishment:62.8	puzzled:36.0	thixotropic:68.8	contraction:27.0
	amazement:53.8	f5:36.0	diana_buds:68.8	goodacre:27.0
	foretell:53.8	glint:36.0	stillstanding_b:68.8	gosh:27.0
neemo	broad:14.1	relative:17.4	:54554.5	30:12.1
	ships:11.0	blah:17.4	simplymayamarie:54554.5	la:9.9
	narrow:10.4	dmc:14.9	chr1smclaughlin:54554.5	nope:7.7
	trunk:9.2	sammy:14.9	glanced:54554.5	tall:7.7
	belonging:9.2	avocado:14.9	shwood:54554.5	ice:7.7

Table 6: Log Odds of words top 5 words in different Datasets