

Emotion Analysis - Assignment 2

Isabelle Mohr (st169866) Nishan Chatterjee (st170030)

January 10, 2021

1 Task Description

The task is to predict emotions of text from two different datasets using two different models. Thus, the first part of the task consisted of making a well-reasoned dataset selection, which will be discussed in Section 2.1. The second part of the task concerns designing two models that predict the emotion labels of the datasets. This process is explained in Section 2.3. A kNN model with handcrafted feature vector was implemented (Section 2.3.1) and a Bert-based model was also implemented (Section 2.3.2). Results are reported in Section 3, reporting globally across both datasets (Section 3.1) as well as per model (Section 3.2). This report ends with a discussion of the outcomes and future ideas in Section 4.

2 Setup

2.1 Dataset Selection

After looking at the datasets available in the unified dataset, we weighed up different aspects of the datasets, looking at their features. We wanted to select two datasets of comparable size, with somewhat comparable emotion labels, with only one label per instance. We decided on the ISEAR dataset and the Tales-Emotion dataset. A brief description can be found below in Table 1.

	ISEAR	Tales
Size	7 665	15 302
Granularity	descriptions	sentences
Topic	events	fairytales
Classes	Plutchik's 8	noemo + Ekman's 6
Labels	1 per inst.	1 per inst.

Table 1: **Dataset Comparison**

The topics of the two datasets are unrelated, which we thought would be an interesting challenge to pose to the two prediction models. Although the emotion classes for the two datasets are not identical, we thought it would still

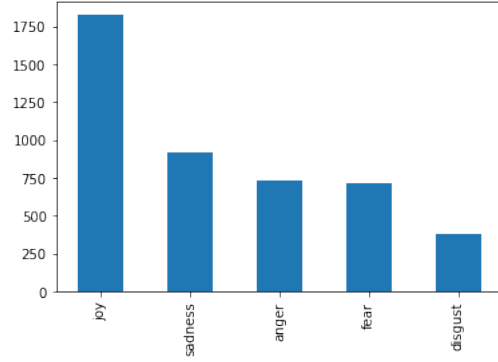
make sense to select the intersection of classes between the two datasets for training and prediction.

2.2 Data Preprocessing

In order to train and predict on our two chosen datasets, we dropped all instances of *noemo* from the data. In Tales, *surprise* was also dropped, while in ISEAR *shame* and *guilt* were dropped. In other words, we kept the intersection of the two datasets in classes, resulting in the classes *anger*, *sadness*, *joy*, *disgust* and *fear*. The distributions of the datasets with respect to instances in each class can be seen in Figure 1b and Figure 1d.

			text
target_emotions	label	data_type	
anger	0	train	622
		test	110
disgust	4	train	321
		test	57
fear	2	train	605
		test	107
joy	1	train	1553
		test	274
sadness	3	train	783
		test	138

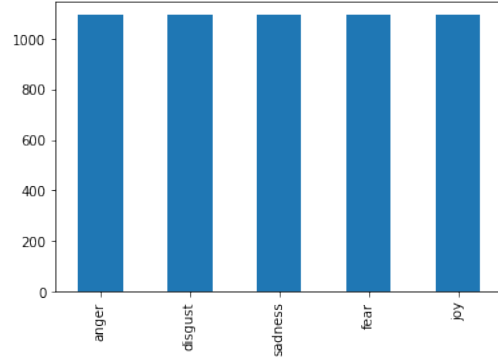
(a) Tales Classes



(b) Tales Class Distribution

			text
target_emotions	label	data_type	
anger	2	train	932
		test	164
disgust	4	train	931
		test	165
fear	1	train	931
		test	164
joy	0	train	930
		test	164
sadness	3	train	931
		test	165

(c) ISEAR Classes



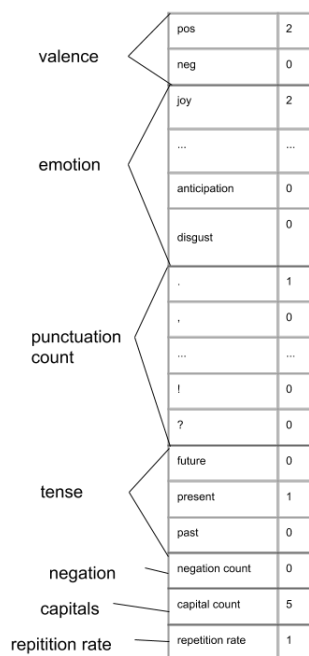
(d) ISEAR Class Distribution

Figure 1: Class distributions of both datasets

2.3 Models

2.3.1 Feature-based Model

We decided to use the kNN algorithm for our feature-based model, with $k = 5$. This is a 'lazy' algorithm that only does one pass over the training data but can do surprisingly well on some kinds of data. Because this model is fast, results are obtained quickly.



valence	pos	2
	neg	0
emotion	joy	2

	anticipation	0
	disgust	0
punctuation count	,	1
	,	0

	!	0
	?	0
tense	future	0
	present	1
	past	0
negation	negation count	0
capitals	capital count	5
repetition rate	repetition rate	1

Figure 2: **Example Feature Vector**

The feature vector was carefully designed using the NRC Emotion Lexicon and a set of other features. For a given instance, each token would be looked up in the emotion lexicon and if an emotion or valence was associated with it, the corresponding position in the feature vector would gain an entry. We also considered features like counts of all punctuation in the text (many exclamation, question marks or fullstops could indicate emotion), the tense of the instance (perhaps there is some correlation between, for example, past tense and *sadness* or future tense and *joy*), counts of negations (*not*, *never*.. may indicate opposites, for example, 'do not like' as anger), as well as counts of capital letters (many capitals could indicate shouting, therefore emotional) and a calculated repetition rate (perhaps repetition indicates some emotion). An example feature vector based on the instance "*I LOVE the sun on my smooth face.*" can be seen in Figure 2.

2.3.2 Deep Learning based Model

The model used here is the Base Bert uncased model which is a pretrained model containing 12-layers, 768-hidden-embeddings, 12-attention-heads, and a 110M parameters. It was trained on lower-cased English text by Google. This is the smaller variant of the model where bert-large-uncased has 336M parameters. The model was chosen as Transformers like Bert have proven to be very successful in understanding nuances in text and sentiment. GPT3 is better than Bert but since availability of the model is an issue, choosing this as a state-of-the-art model to compare with our handcrafted feature representation seemed like a good idea.

3 Results

3.1 Global Results

Our results turned out mostly as expected, with the Bert-based model outperforming the kNN model by a large margin. The average accuracy scores (%) for the kNN model and the Bert-based model can be found in Figure 3a and Figure 3b respectively.

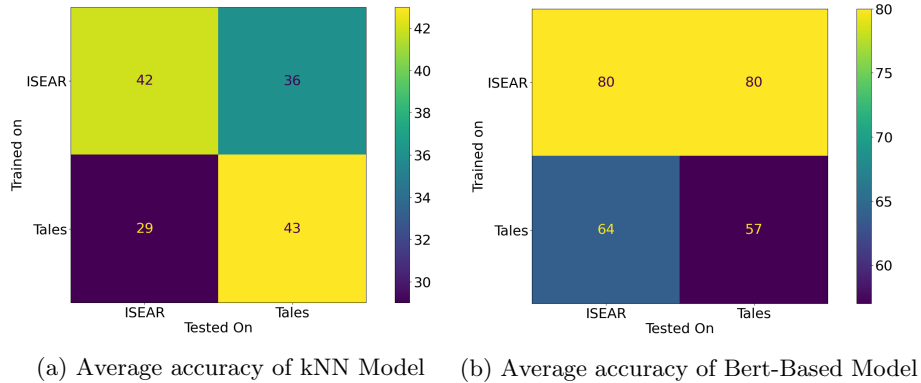


Figure 3: Average accuracy (%) across two models

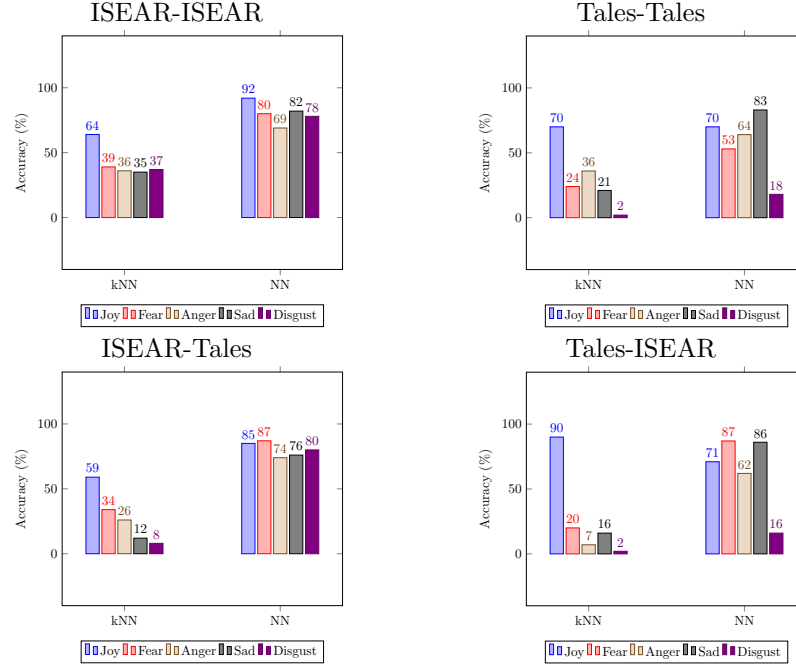
These global results show that with the kNN model, we were able to achieve 42% and 43% accuracy when training and testing consistently with either the ISEAR or Tales dataset. Accuracy scores with the kNN model are lower when testing on the corresponding alternate dataset, which is expected because these datasets do differ in topic. Thus, when trained on ISEAR, predictions on Tales only achieve 36% accuracy, and when trained on Tales, predictions on ISEAR are even lower at 29% accuracy.

The train-test split was 85-15%. But it should be noted that in our implementation, while evaluating a model trained on a dataset (tales-emotions for

example) and evaluating it on another dataset (in this case, ISEAR), the test part of the code automatically becomes the validation set as it's unseen data.

However, the Bert-based model scores look slightly different :)

3.2 Results per class



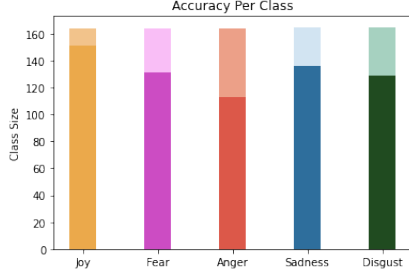
The figures above show the accuracy (%) scores achieved by both models on the different emotion classes. The Base-Bert model (here NN) consistently performs better across all classes than the kNN with feature vector. The worst performing class for both models is *disgust*, for which there are very few instances in the Tales dataset. However, even when trained on ISEAR, kNN is not able to predict *disgust* instances in the Tales dataset. This could be because instances of *disgust* look very different in the Tales dataset to those in the ISEAR dataset. The kNN model also consistently predicts instances of *joy* best across both datasets. Perhaps the feature vector that was handcrafted captures features of *joy* better than other classes.

3.3 Results for Base-Bert

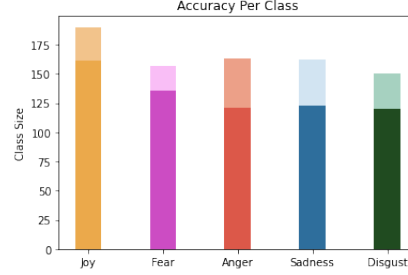
Ideally, the model should have been run with multiple random seed initialization and trained for 10-15 epochs to see where the performance peaked and the model started over-fitting. But since we trained our model on Google Colab which comes with smaller server times, the results on how the model scales with more training have not been included.

However, this model clearly indicates how using pretrained models and fine-tuning them on custom datasets can still achieve significantly higher performances in comparison to a hand-crafted model where context or word sense is not fully captured.

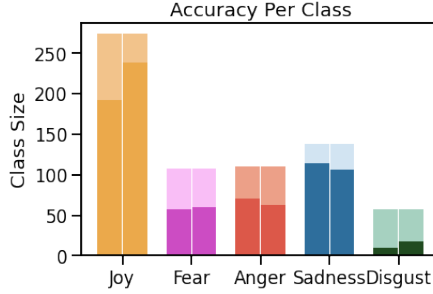
The Bert model also indicates that disparity on class sizes can lead to performance gaps as shown in the tales-vs-ales confusion matrix.



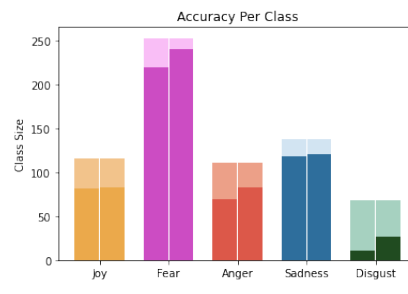
(a) Accuracy per class for ISEAR-ISEAR



(b) Accuracy per class for ISEAR-Tales



(c) Accuracy per class for Tales-Tales



(d) Accuracy per class for Tales-ISEAR

4 Discussion

As mentioned in Section 2.1, the datasets were mainly chosen based on easy class comparability. Taking two datasets from different topics posed a good challenge to our models. Taking only the intersection of classes between the two datasets means that unfortunately the already small Tales dataset was reduced a little more. The ISEAR dataset still had a largely normal distribution of instances across classes. We originally wanted to use the Blogs dataset instead of the ISEAR dataset but could not be granted access in time.

The class imbalance in the Tales dataset may be the reason both models get lower scores when training on Tales. This effect is most clearly seen in the Base-Bert model in Figure 3b. However, an increase in Base-Bert model performance can be seen over multiple epochs, particularly when training on Tales as described in Section 3.3. In future, the model should be run for longer if computational resources allow, possibly yielding better results.

As for the kNN model, further developments could be made with regard to the feature vector, by extending it, for example, with tf.idf scores. However, we think the kNN results are a good starting point given this hand-crafted feature vector, though deep learning methods have surely surpassed these very creative efforts and designs by a long way and will continue to do so.

The training for the base-bert model was intended to be implemented for only 3 epochs as state of the art performance was not really the goal but rather to check how the model compared to a handcrafted model and how the two datasets compared to each other and why. Further analysis is yet to be done in order to understand the nuances of each of the datasets used and visualization techniques like attention analysis for performance comparison or plotting the emotion space and looking at outliers may have been more effective.

PS: Thanks for reading and we hope you have a lovely year ahead. :)