Emotion detection in social media data

University of Pisa
Human Language Technology
Project report

Andrea Alberti
a.alberti14@studenti.unipi.it
Roll number: 588945

Alberto Dicembre
a.dicembre@studenti.unipi.it
Roll number: 668377

Giuseppe De Marco
g.demarco5@studenti.unipi.it
Roll number: 581658

May 29, 2024

1 Introduction

The goal of our project is to carry out an emotion detection task over text data extracted from social media. Emotion detection tasks have a variety of usages, like market research, the possibility of writing software able to change its behavior according to users' feelings, or cyberbullying victim detection. Using two datasets and different models, our work focused mainly on the comparison of their performances, to determine which hyperparameters and architectural choices worked best. The analysis was centered around: 3 *encoder-only* transformer models (**BERT** [1], **RoBERTa** [2] and **SocBERT** [3]), 3 *bag-of-words* based classifiers (**Naive Bayes**, **decision tree** and **random forest**) and a recent *decoder-only* model as zero-shot and few-shot classifier: Meta's **Llama3**.

2 Datasets

As mentioned, we used two datasets: **GoEmotions** [4], based on Reddit comments, and a Twitter dataset [5] that comes with no name and will be referenced in this report as **TwitterData**.

2.1 GoEmotions

GoEmotions [4] is the largest human annotated (with multiple raters) dataset of **58k Reddit comments**, extracted from popular English subreddits. It comprises **27 emotions** (plus a *neutral* one) and it's designed for **multilabel classification**. The dataset already comes with an *hold out* split version. The authors designed their own taxonomy to provide a more fine-grained emotion expression, but have also included a mapping of the labels according to *Ekman's* taxonomy. Our studies took into consideration both the dataset with standard labels and their grouped version according to the mapping.

2.2 TwitterData

TwitterData [5] is a dataset of **20k tweets** suited for **single label emotion detection** tasks based on *Parrott's* taxonomy. The dataset is already split in *hold out* fashion.

3 Prior work / Related work / Literature review

Out of the two datasets, only GoEmotions comes with a paper, which describes its characteristics and performs an analysis on it using **BERT**, both with base labels and grouped ones. This study shows that even humans classify fine grained emotions with high variance due to subjectivity of emotions definitions.

Additionally, referring [6], we discovered that emotion detection on tweets is easier than on Reddit posts due to differences in language (i.e. emoji usage) and in records nature (tweets are more self contained).

Most papers tackling this task analyze transformer models and use macro average f1 score as main evaluation metric. The most used transformer, either as a baseline to compare with newer transformer models or to analyze improvements with respect to well known models such as Naive Bayes, is BERT. The authors of GoEmotions analyzed this model with such score in [4] and an additional study on this dataset (see [7]) also uses BERT with some additional pooling techniques on its output.

TwitterData is the less famous dataset of the two, but many studies are present on it. A relevant one using RoBERTa can be checked in [8], where a lot of data preprocessing was performed and only micro averaged f1 score was examined, reaching 0.927.

4 Data exploration

We analyzed the distribution of gold labels and, as Figure 2 shows, there is high imbalance in both datasets (more evident in GoEmotions). This distribution is the same across the hold out split implying **stratification** was performed by the authors of the datasets. By using regular expressions we found no links in GoEmotions records, 58 user tags and 194 subreddit tags.

In TwitterData, no *hashtags* or *mentions* are present. The dataset authors already performed some **text normalization** on records: punctuation is removed and *clitics* are not expanded but attached to the main word. Despite this initial cleaning, hundreds of records contain tokens that are related to presence of images or links and various *metadata*. These tokens tend to appear clustered at the end of records rather than mixed with meaningful words. There is no easy way of detecting links in this dataset due to high inconsistency in how they are split into tokens after the authors cleaning.

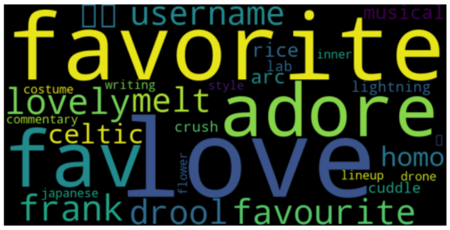
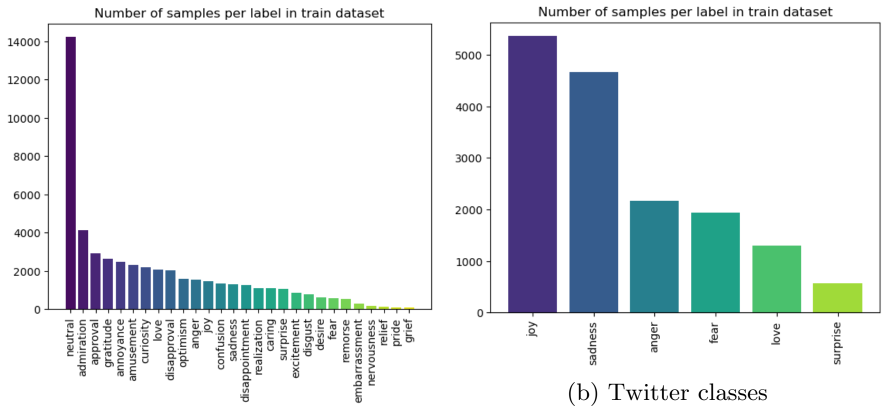


Figure 1: The image shows the top 30 words according to PMI with a frequency cutoff threshold of 10 for the emotion *love* of GoEmotions. The two unrendered words are emojis: one is a red heart and the other is a face with hearts as eyes. Some irrelevant tokens are also shown such as *lab*. This is one of the few cases where useful emojis are considered relevant after using the frequency threshold: many emoticons are present in this dataset, but most of them are not repeated over many records. An interesting trade off: a lower threshold detects many emojis, but also many irrelevant rare tokens.

In order to understand space and time performances of the three transformers fine-tuning, we analyzed the distribution of tokens according to the tokenizers provided with the models. TwitterData has, for every split and tok-



(a) GoEmotions classes

(b) Twitter classes

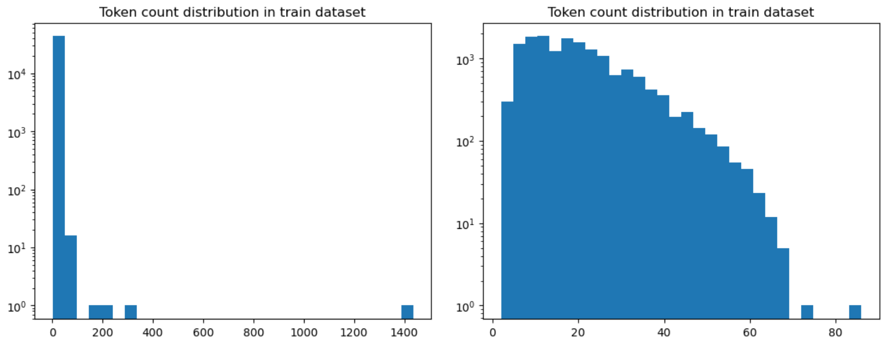
Figure 2: Distributions of classes in training split of the two analyzed datasets showing high class imbalance. The distribution is the same also for validation and test sets.

enizer, a **low variance distribution** of tokens with numbers never exceeding **90**. GoEmotions, instead, has some remarkable **outliers** only in the training set, reaching around **300 tokens** with *Bert Tokenizer* and **1400** with the others. This is more than the *input size* of the models, **truncation** was used to address this when fitting them on the unpreprocessed version of the datasets (see Section 5). A visual example of the results can be seen in Figure 3. We found out that these outliers are correlated, with long sequences of repeated characters (even more than 10): only very few records contain these sequences and almost every outlier contains one. Also a couple of records among these contained an *ASCII image*.

5 Data preparation

In GoEmotions, we substituted **user tags** with '[NAME]': this is the same token used by the dataset authors to hide people nouns for privacy reasons, our choice derives from the fact that user tags also refer persons, so this substitution should be inline with the original preprocessing. **Links** have been substituted with '[LINK]' since their names are expected not to be useful for our task. The long *ASCII image* present in the dataset was removed from the record.

In TwitterData we chopped the final parts of records containing **metadata tokens**. This was done considering tokens that were never mixed in the good portion of the tweets (thus privileging *precision* with respect to *recall* in order to avoid potential loss of good information in many tweets by cleaning relatively few dirty records). **Links** where converted to the token '[LINK]' by using a *regex* 'fit' on this dataset.



(a) GoEmotions tokens

(b) Twitter tokens

Figure 3: Distributions of tokens in training split of the two analyzed datasets according to RoBERTa tokenizer. On GoEmotions there are very evident outliers, while in TwitterData the distribution has a much lower variance.

In both datasets, **long sequences** of the same character have been capped to 5. All of these modifications only affected a tiny portion of the datasets, so we don't expect a radical change in the performances of our models. A **cleaning validation** regarding *data preparation* will still be discussed in Section 8.1. In any case, in both datasets, no records were dropped.

6 Models

Our **baseline** is given by Naive Bayes models (*Multinomial* based and *Bernoulli* based), decision trees and random forests. All of them where trained with *scikit-learn pipelines* on a *tf-idf* weighted count matrix; further details on the types of pipelines used are discussed in Section 7. The **tokenizer** used to get frequencies is *NLTK punkt*.

All **transformer models** (BERT, RoBERTa, SocBERT and Llama3) were accessed through HuggingFace's transformers library. The former three have been adapted from the base pre-trained versions adding a **head** consisting in a *dropout* layer (with 0.1 coefficient), followed by a *sigmoidal* output layer. This layer was chosen to be able to address the **multilabel** classification task but the output will still be interpreted depending on the dataset (TwitterData as **single label** by only considering the *argmax*, GoEmotions considering all probabilities above a given *threshold*). Concerning Llama3, the 8 billion parameters Instruct version has been used, in conjunction with the *outlines* library to provide a **structured** output generation.

7 Experiments

Every architecture considered was used to get three models: fit on TwitterData, GoEmotions base dataset and a variant of it, where labels are grouped to provide *Ekman*'s taxonomy according to the mapping used in [4].

Both datasets are provided with an *hold out* split in which the validation set was used both to find good **hyperparameters** for our models and to find the best data cleaning technique for each model. This includes **validating** our data preparation approach, by fitting models on the *unpreprocessed* and *preprocessed* versions of the datasets using the same ***seed***.

7.1 Baseline

In all *bag-of-words* approaches we used **tf-idf** score matrices and considered:

- removing **stopwords**;
- using a minimum **df threshold**;
- performing *feature selection* with **PMI score**;
- normalizing text by performing **lemmatization** (with *pos tagging* to improve performances);
- using **one-vs-rest** also for TwitterData.

For the Naive Bayes models we also considered comparing *Multinomial* Naive Bayes with *Bernoulli* Naive Bayes and changing the *add-k* **smoothing** parameter.

Specifically for decision trees and random forests we attempted to tune the **minimum samples** needed to split.

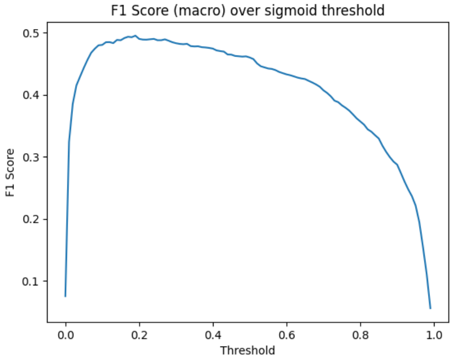


Figure 4: The image shows the Macro-averaged F1 score obtained for different thresholds applied to the same prediction probabilities. In particular, one can see that simply accepting the default 0.5 threshold would yield worse performances.

For random forests also the **number of features** used by each tree was tuned.

7.2 Transformers

The hyperparameters for BERT on GoEmotions have been taken from [4]. For the other models to fine-tune, a small *grid search* was performed to detect changes of performances changing **batch size**, **learning rate** and **weight decay**. For multilabel classification we also considered tuning the **threshold** to detect the presence of emotions given the *sigmoidal* output layer predictions according to the *macro-average* F1 score.

For RoBERTa we also considered **freezing** some layers in order to speed up the fitting process.

In each fit (inside and outside *cross validation*), a number of *epochs* big enough not to *underfit* was used (considering both papers and preliminary learning curves, in any case never more than 6) and at the end of the training, the weights of the *epoch* with best results over the validation set are restored.

With Llama3, the inference process was aided by the **outlines** library, which forced the generation process to choose from a set of choices. In the single label case, the name of the single class was directly inferred; in the multilabel one, the process consisted in asking the model to assert the manifestation of each single emotion, for every sentence. This was tested with both a **zero-shot** prompt, and a **three-shot** one, using examples from the training data.

7.3 Scores

macro-average F1 score is the metric that received the greatest attention since it's needed to compare our results with most papers (not all of them since some used the **Micro-average** instead). When evaluating results on GoEmotions we also considered the **Jaccard** score with *macro* and *samples* average to have a more flexible metric with respect to *accuracy.*

7.4 Mapping analysis

Other than following the **Ekman mapping** on GoEmotions, provided by the dataset authors in [4] to fit a model on grouped labels, we studied mapping induced by training on the dataset with original classes and then comparing the predictions with the **grouped ones**. This allows to understand how the models would have mapped the labels without forcing them to do so. The resulting **normalized confusion matrix** values are *dense* vectors representing similarity of each of the original emotions with respect to every grouped one. The resulting vectors could thus be interpreted as an embedding of the whole original emotion on the dimensions of the grouped ones.

The same procedure was also done to study a mapping **between emotions** of the two datasets. In this case the mapping was defined by us and the results are also meaningful to understand performances in case of a *domain shift.*

7.5 Word rankings

An additional analysis regards ranking words of each emotion to get its most representative words. This allowed us to get additional insights on effects of class imbalance and limits of our approaches. The ranking was performed using **PMI** after applying **smoothing** and a minimum **df threshold** to the word counts.

This addressed the main score problem of detecting irrelevant rare features as highly important.

8 Results and Analysis

8.1 Cleaning validation

Final transformer models were all fit on the unpreprocessed datasets since our data preparation was too minimal to make a difference. There is not a clear winner between the original and cleaned datasets since the performance differences were more dependent on the **weight initialization** (seed) than on the cleaning itself. The **fluctuations** observed by using the same seed were of 0.01 at most, while the variance produced by changing seed caused instead differences of even 0.06. Figure 5 shows a comparison between a RoBERTa model fit on the cleaned version of GoEmotions and the uncleaned one: despite some small differences, they have very similar macro average f1 scores.

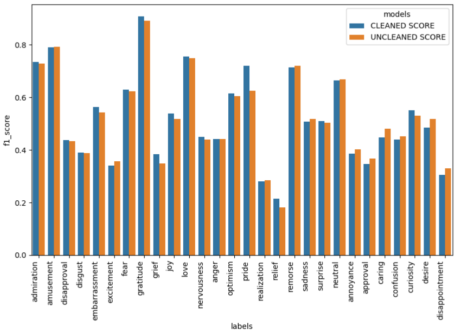


Figure 5: The image shows the F1 score obtained on GoEmotions validation set with and without performing data preparation. Despite small changes in various classes, the macro average of the cleaned version is 0.004 higher than the unpreprocessed one. This result highly depends on the seed used for comparison.

Regarding baseline models, we still fit the final models on unpreprocessed datasets since scores were consistently the same with or without our data preparation. It is worth mentioning that additional text normalization techniques like lemmatization are discouraged since they consistently yielded slightly worse results.

8.2 Best models

In this section, we provide the best settings for each model to allow reproduction of results. Parameters not present here are either constant values discussed elsewhere in this paper or default library values (of scikit-learn for baseline models and pytorch for transformers).

For GoEmotions:

- Bayes: One vs rest, Bernoulli Naive Bayes, 5 minimum token frequency, feature selection with PMI and 1000 features, td-idf score correction, 0.01 for add-k smoothing

- Decision Tree: One vs rest, tf-idf score correction
- Random Forest: One vs rest, tf-idf score correction, per tree features: 2000, minimum samples to split: 10
- BERT: max epochs: 4, batch size: 16, learning rate: 5e-05, weight decay: 0
- RoBERTa: first 9 RoBERTa layers frozen, max epochs: 6, batch size: 32, learning rate: 5e-05, weight decay: 1e-05
- SocBERT: max epochs: 6, batch size: 16, learning rate: 5e-05, weight decay: 0
- Llama3: outlines and True or false prompt

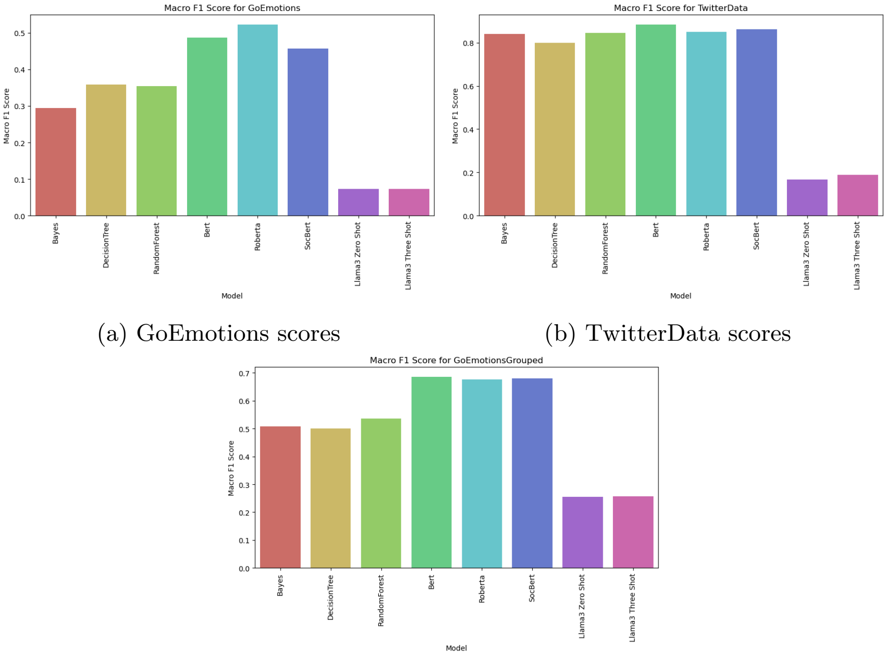
For TwitterData:

- Bayes: One vs rest, Bernoulli Naive Bayes, 5 minimum token frequency, feature selection with PMI and 300 features, td-idf score correction, 0.01 for add-k smoothing
- Decision Tree: tf-idf score correction
- Random Forest: One vs rest, tf-idf score correction
- BERT: max epochs: 6, batch size: 16, learning rate: 5e-05, weight decay: 1e-05
- RoBERTa: first 9 RoBERTa layers frozen, max epochs: 6, batch size: 32, learning rate: 5e-05, weight decay: 1e-05
- SocBERT: max epochs: 6, batch size: 32, learning rate: 1e-05, weight decay: 1e-05
- Llama3: outlines with choice prompt

8.3 Performance comparison

Every analyzed model suffers from class imbalance. In GoEmotions the trend is to have higher scores for bigger classes and a very low score (sometimes even zero) on the least expressed ones (like grief and relief). This can be seen from Figure 6. Additionally, models predictions are biased toward bigger classes, such as 'neutral' (the biggest of all) and, when this model is tested on TwitterData, no records are predicted as 'grief' (one of the smallest classes), as shown in Figure 8.

Figure 7 shows the comparison of the models on the test set according to **macro-averaged F1**. Plots comparing models according to Jaccard are omitted since they are very similar to the F1 ones.



(a) GoEmotions scores

(b) TwitterData scores

(c) Grouped GoEmotions scores

Figure 7: Final models performance on test set of each dataset according to macro average f1 score.

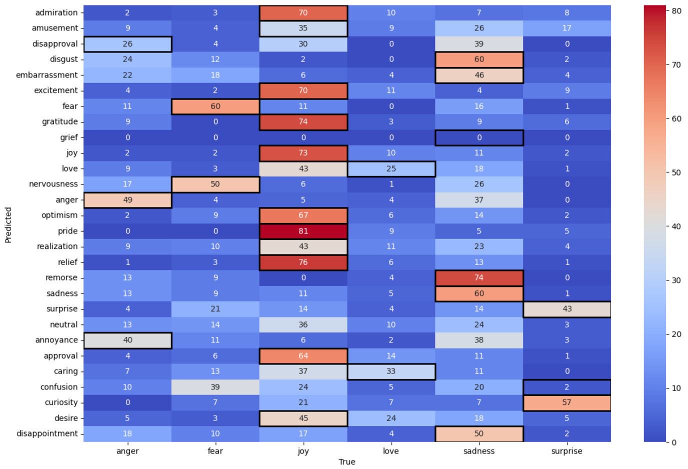


Figure 8: The image shows the confusion matrix of BERT predictions on TwitterData after being fit on GoEmotions. Counts are normalized per row to provide a better visualization. The black border around the cells highlights the true correspondence according to our proposed mapping.

One can notice a meaningful increase of performances going from baseline models to fine-tuned transformers. This is more evident on GoEmotions. Llama3 is below the baseline: in multilabel classification tasks it detected almost every emotion in all records, thus producing a non meaningful classification. This model achieved better results on TwitterData, but it still had the lowest scores since it only discriminates decently two out of six classes. Moving from zero-shot to three-shot classification did not improve results on GoEmotions and slightly increased performances on TwitterData.

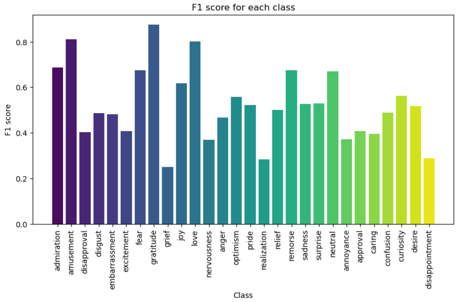


Figure 6: The image shows the F1 score obtained on GoEmotions test set with RoBERTa, the best model for this dataset.

Another evident trend is that scores among models in GoEmotions tend to be lower and with an higher **variance** with respect to TwitterData. This is likely due to:

- TwitterData records being more **heavily preprocessed** by the dataset authors, reducing a lot of linguistic variance;
- GoEmotions having many more classes with more **label imbalance**, thus hampering scores due to classes with very low *support* being not well predicted;
- tweets being **easier** to be classified in general, as detailed in [6].

The performances of the best model of each dataset (the two base ones plus the grouped GoEmotions) have been also statistically validated with all the rest by performing bootstrap testing. This helped in understanding which differences among the models are significative. None of Bert, RoBERTa and SocBERT are meaningfully better among themselves both in the grouped GoEmotion dataset and TwitterData. On the latter, BERT isn't even statistically better than Bayes (0.15 p-value): it is also evident from the comparison plot that performances are very close across models. On the base version of GoEmotions, all p-values are very low (less than 0.04), implying statistical meaningfulness of the differences among the models.

One interesting observation regards performance variation over models that participated to the same **grid search**: RoBERTA models were less sensible to initialization and usage of different hyperparameters (only producing differences of around 0.04) than BERT and SocBERT (where changing hyperparameters affected the scores of even 0.3).

8.4 Comparison with papers

Even by following the hyperparameters in [4] for BERT on GoEmotions, the final macro F1 score of 0.46 was not reproduced and, instead, our model scored 0.48. The same result was observed in [7], where the authors of the paper attributed this discrepancy to their different **pooling** technique. In our case, the increase is given by **threshold tuning**, as shown in Figure 4. More in general, threshold tuning proved useful in increasing performances of all of our transformer models.

The amount of **epochs** needed to learn the task without *overfitting* was mostly around 2-3 epochs (depending on the model and dataset). This is, for GoEmotions, less than the amount proposed by the paper authors (that is 4).

In the grouped version of GoEmotions, scores are higher than with the base dataset due to less class imbalance. Interestingly, the performances obtained by our fine-tuned transformers by simply grouping predictions (without explicitly forcing the model to learn grouped labels) are only slightly worse than fitting the models after mapping the labels. For example, BERT achieved 0.64 macro averaged f1 (same performance of the authors of GoEmotions) when mapping default predictions and 0.68 by training on grouped labels.

Considering the study [8], we obtained the same accuracy results with RoBERTa even without complex data preprocessing and BERT has a slightly higher score: 0.93.

8.5 Mapping analysis

The **Ekman mapping** of GoEmotions labels was analyzed by predicting grouped labels with models fit on the original classes. There is a tendency to map records to *'joy'*, this happens because some of the biggest classes are grouped under it. Baseline models have some problems in mapping the smallest classes. Considering transformers, instead, there is an agreement among the models and the resulting mapping is also what we expected. However, this doesn't coincide with the mapping proposed in the original paper, where the class *'realization'* is mapped to *'surprise'*: both the transformer models and us would have instead mapped it to *'joy'*.

Regarding the mapping with **Parrott labels**, the analysis was performed by testing models fit on GoEmotions on TwitterData. A visive example of the induced mapping can be seen in Figure 8. Due to the heavy domain shift, baseline models did not guess many emotion mappings. Transformer models, instead, agree again on the emotion mapping and, for most emotions, the results coincide with our *'gold mapping'*. We noticed that no record is predicted as *'relief'* (one of the small classes), thus not inducing any mapping choice for this label. Emotions connected with Parrott's *'love'* also tend to be mistaken for *'joy'*. If we consider the **domain shift** and how close this results are with respect to our mapping, we can conclude that these models have good generalization capabilities, not in the sense that they would beat models trained directly on TwitterData, but that they learned some semantical aspect of emotions that they can use when changing domain.

8.6 Word analysis

Figure 1 shows the top 30 relevant words for class *love* in GoEmotions. After a few attempts trying to generate word rankings for each emotion, it was found that a **df threshold** of 10 in GoEmotions and 5 in TwitterData was important to prevent too much importance to be given to very rare but non meaningful words. However, we noticed that some tokens, especially **emoticons**, are relevant to easily detect the emotions in a text, but they are also so rare that it's hard if not impossible to do statistically meaningful inference using them. These useful but rare features showcase an evident limitation of our approaches, especially when considering fitting baseline models, that do not exploit *transfer learning*. Indeed, by ranking the best features considered by the baseline and comparing them with the models agnostic rankings, we can notice various similarities in words and very rare tokens and emojis are indeed not among the top features considered.

9 Conclusions

In conclusion, our work has shown that, as expected, increasing granularity (i.e. number of classes) in an emotion classification task makes it way more difficult, as definition of emotions is highly **subjective**. Moreover, models performances on TwitterData were higher presumably because of less class imbalance and a more heavily cleaned data. In addition, as we have seen in Section 8.5, models trained on GoEmotions expressed a good generalization capability when transferred to the TwitterData task.

Possible enhancements might include:

- a more extensive **grid search**: since a lot of time is needed for each model fit, only three hyperparameters were tested and with very few values. It would be interesting to also consider using different classification *heads*;
- the inclusion of **explainability studies** on the transformer models, with the goal of understanding possible nature of errors;
- more extensive tests on **Llama3**, perhaps using the 70 billions parameters version, or using different methods for dealing with the structure of generated text, other than using *outlines* library.

References

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[3] Y. Guo and A. Sarker, "SocBERT: A Pretrained Model for Social Media Text," in *Proceedings of the Fourth Workshop on Insights from Negative Results in NLP*, 2023.

[4] D. Demszky, D. Movshovitz-Attias, J. Ko, A. S. Cowen, G. Nemade, and S. Ravi, "Goemotions: A dataset of fine-grained emotions," *CoRR*, vol. abs/2005.00547, 2020. [Online]. Available: https://arxiv.org/abs/2005.00547

[5] PRAVEEN, "Emotions dataset for nlp classification tasks." [Online]. Available: https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp

[6] E. Öhman, M. Pàmies, K. Kajava, and J. Tiedemann, "Xed: A multilingual dataset for sentiment analysis and emotion detection," *arXiv preprint arXiv:2011.01612*, 2020.

[7] N. Alvarez-Gonzalez, A. Kaltenbrunner, and V. Gómez, "Uncovering the limits of text-based emotion detection," *CoRR*, vol. abs/2109.01900, 2021. [Online]. Available: https://arxiv.org/abs/2109.01900

[8] TAKAI380, "Emotion detection from tweets: Roberta fine 8eda50." [Online]. Available: https://www.kaggle.com/code/takai380/emotion-detection-from-tweets-roberta-fine-8eda50/notebook