Emotion Hunters at EMit: Categorical Emotion Detection combining BERT and ChatGPT models

Gianluca Calò[1], Francesco Massafra[1], Berardina De Carolis[1] and Corrado Loglisci[1]

[1]Department of Computer Science, University of Bari, Italy

Abstract

Emotion detection in text plays a crucial role in various applications, such as customer feedback analysis, social media monitoring, or for the analysis of the verbal part of human communication. Deep learning techniques have shown promising results in accurately recognizing and classifying emotions in textual data. This paper describes the approach to categorical emotion detection of the Emotion Hunters team. After a preprocessing phase, a model fine-tuned from AlBERTo together with the ChatGPT APIs was used to address the challenge. The results show that on the out-of-domain test set our approach performed better than on the in-domain one thus showing a good generalization capability.

Keywords

Emotion Detection, BERT, ChatGPT

1. Introduction

Emotion detection in texts has gained significant importance in recent years due to the pervasive presence of digital communication platforms and the wealth of user-generated content. The ability of software to understand and analyze emotions expressed in a text has numerous applications across various domains, including marketing, customer service, mental health, and social sciences. Emotion detection involves the use of Natural Language Processing (NLP) techniques and machine learning algorithms to automatically identify and classify emotions conveyed in textual content. By accurately detecting and interpreting emotions, we can gain a deeper understanding of human experiences, opinions, and attitudes.

As far as emotion detection for the Italian language is concerned, it presents distinctive features which range from morphological to lexical viewpoints. It presents a lot of words with particles in two or three units (e.g., verbal groups), which are difficult to label. The words used to express the same idea can have different types of grammatical categories, they can be nouns and verbs. In addition, they can be associated with general semantic categories or specific categories. Italian is a very rich language with words that hold more than one meaning, which may mislead an automatic emotion detector. Moreover, while many linguistic resources and annotated texts have been generated for wide-coverage languages, such as English, Chinese and Arabic, the same cannot be said for other less-resourced Indo-European languages, such as Italian.

The EMit (Emotions in Italian) Subtask A [1] at EVALITA 2023 [2] aims at detecting emotions in social media messages about TV shows, music videos and advertisements. Given a message, the system has to decide which emotions are expressed in the message or if the message is a neutral one. According to the annotation of the dataset, the problem to address is designed as a multilabel classification one. Therefore, the system, given a message, will classify it and return all the possible labels denoting emotions contained in it. In particular, in Subtask A, the message could be classified as neutral, or expressing one or more emotions in the following set of 10 labels: anger, anticipation, disgust, fear, joy, sadness, surprise, trust, neutral [3], and the additional label love. Our team, the EmotionHunters, addressed this challenge with a two-steps model. After a pre-processing phase, we fine-tuned a model based on *AlBERTo* [4] and test it on the validation set. The performance on the validation set exceeded the baseline of about 16% reaching an accuracy calculated with the weighted F1-score of 0.56. However, when we run the model on the provided in-domain test-set, we noticed that in some cases the model did not make predictions and that there was a high number of neutral predictions on the total of the results. Then, since the beginning of the call for this challenge, ChatGPT became very popular, for these two cases, we integrated the ChatGPT APIs 3.5 [5] and this increased the prediction of the model of 1%. The proposed model has been tested on the two test sets proposed by the challenge: in-domain, including tweets of the same textual genre and subjects of the training set, and another one, out-of-domain, including social data of different genres and subjects. Our approach showed to have a better performance on the out-of-domain test set showing that it is able to generalize with respect

EVALITA 2023: 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Sep 7 – 8, Parma, IT

g.calo26@studenti.uniba.it (G. Calò);
f.massafra7@studenti.uniba.it (F. Massafra);
berardina.decarolis@uniba.it (B. D. Carolis);
corrado.loglisci@uniba.it (C. Loglisci)

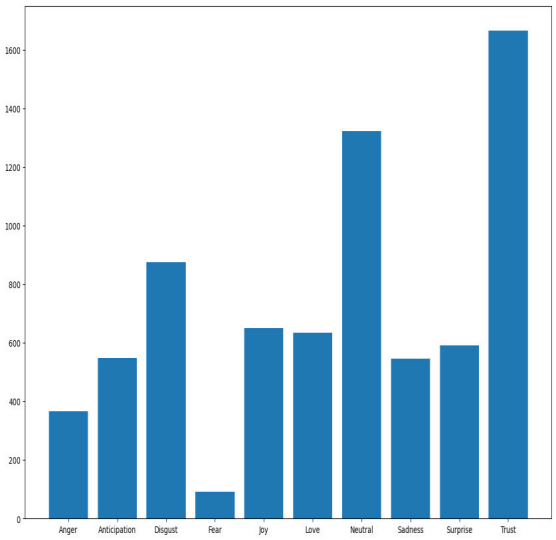

Figure 1: Distribution of examples for each emotion.

to the topic. This is for us an interesting result because we want to apply the model in different domains like the one of conversational experiences with intelligent agents. We did not have the time within the challenge deadline to train a model based only of LLMs like ChatGPT and this is part of our future work.

2. Description of the system

In the following, we first describe the pipeline of text pre-processing and then provide details on the classification algorithms used and configured for the task at hand.

2.1. Analysis of the Dataset

The provided training dataset consists of a collection of 5966 labeled tweets, each identified with one or more labels related to the predicted emotions (among the 10 emotions mentioned above) (see Figure 2). The class distribution is not homogeneous, with the trust and neutral classes being predominant, while the rear class is the least frequent (see Figure 1).

To augment the number of sentences of the fear class, we integrated the training dataset with sentences taken by the dataset MultiEmotions-It [6], moreover, using the affective dictionary proposed in [7], we changed affective terms with synonyms in this way the size of the fear class was upsampled to 400 sentences.

2.2. Pre-processing

A pipeline of preliminary operations is performed in order to first clean and standardize the input messages and then prepare them for a format suitable to the selected learning algorithms [8]. More precisely, we remove the symbols used in social media communication (emoticons, url, links, mentions) without discarding the relative contents but we assign them to semantic categories denoted as tags. For instance, the mentions "@NickName" are converted into the tokens with the uniform and generic tag $< user >$. Each emoticon is converted into the textual description of its meaning taken from a predefined collection of emoticon-description pairs we made for the purpose of this work. For instance, the emoticon with grinning face with big eyes would be converted into the description *<faccina con un gran sorriso e occhi spalancati>* (in Italian). Also, words with hashtags are split into the single tokens, which are then reported with the open and close tags $< hashtag >$ and $< /hashtag >$. For instance, the hashtag "#Sanremo2020" would be converted into the tagged sequence $< hashtag >$ *Sanremo 2020* $< /hashtag >$ composed of two single tokens. The rationale behind these operations is to make tokens and symbols expressing emotions explicit and jointly to augment the features describing the original text. So, the learning process can work on multiple sources of information and better capture the emotive content.

Next, we perform a conversion operation to represent the tweets pre-processed in the input format for the selected learning algorithms, that is, BERT models and variants (as explained in the following). All the tokens

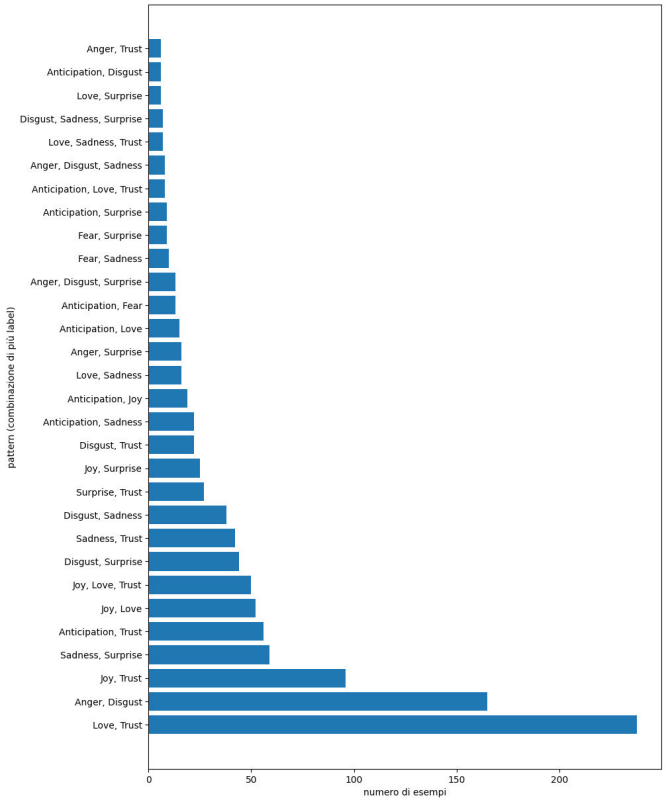

Figure 2: Pattern of labels.

produced for the pre-processed tweets were indexed and used to create a dictionary for the input vectors to the learning process. Considering the typical length of the tweet, usually very less the maximum number of admissible digits, we chose an input length of 128 (elements of the vectors) and prepared an attention mask to decrease the importance of the elements inserted into the input data for padding. Each input vector is in binary code and each element represents the presence/absence of the corresponding indexed token.

2.3. Selected Models

The experimentation is based on BERT models (and variants), which have achieved state-of-the-art results in text classification tasks. BERT utilizes special tokens [CLS] and [SEP] to indicate the beginning of the input sequence and the separation between sentences.

In this specific case, the contextualized embedding associated with the [CLS] token is used as the embedding for the entire sentence. Thanks to the multi-head attention mechanism, it can capture the semantics of the entire sentence effectively.

Several instances of pre-trained BERT models that contain at least some Italian text in their training corpus were considered. For each model, a fully connected layer was added to perform multi-label classification and fine-tune the models on the specific dataset. The models considered are:

- dbmdz/bert-base-italian-xxl-cased [9]: The MDZ Digital Library team released "Italian BERT cased XXL," a BERT version pre-trained on two corpora. The first corpus consists of texts obtained from a Wikipedia dump and various texts from the OPUS collection (http://opus.nlpl.eu/), with a total corpus size of approximately 13 GB and over 2 billion tokens. The second corpus is the Italian part of the OSCAR corpus (https://traces1.inria.fr/oscar/), with a final corpus size of approximately 81 GB and over 13 billion tokens. The "cased" version was chosen as it aligns better with the chosen pre-processing method, as previously done in [8].
- AlBERTo-Base, Italian Twitter lower-cased [4]: A BERT model trained on a corpus of 200 million Italian tweets.
- UmBERTo-Commoncrawl-Cased [10]: A RoBERTa model (a variant of BERT) trained on an Italian sub-

corpus of OSCAR as the training set. It uses a ten-fold version of the Italian corpus, which consists of 70 GB of raw text data, 210 million sentences, and 11 billion words. The sentences were filtered, shuffled at the line level, and utilized for NLP research.

- MilaNLProc/feel-it-italian-emotion [11]: An adapted version of UmBERTo for classification on the Feel-IT dataset.
- bert-base-multilingual-uncased [12]: A multilingual uncased BERT model.

2.4. Training the Models

The challenge provides two baselines and the corresponding code to reproduce their execution. The first baseline uses count vectors to represent the documents based on token frequency, while the second baseline uses TF-IDF vectors. Both implementations limit the vector dimensions to 5000. In the emotion recognition task, the TF-IDF baseline performs better, achieving an F1 score of 41.48%.

For both the baselines and the experiments conducted in this work, a seed was fixed to ensure reproducibility.

Taking into consideration the work presented in GoEmotions [13], we decided to freeze the layers of the pre-trained BERT model and train only the additional classification layers.

Various preliminary experiments were conducted by manually modifying the model's hyperparameters, such as the number of epochs, batch size, learning rate, etc., to understand which was the better approach for this task. To improve the results, the following decisions were made: The Optuna library was adopted to systematically test different combinations of hyperparameters. The MLFlow library was used to track intermediate (F1) and final results (F1 and metrics for each class).

The hyperparameter search space was defined as follows:

- Learning rate: between 2e-05 and 5e-05
- Epsilon (AdamW optimizer): between 1e-8 and 1e-6
- Hidden dropout probability: between 0.1 and 0.3
- Patience for early stopping: a discrete interval between 1 and 5
- Batch size: 16, 32, and 64

The numerous trials conducted using Optuna allowed us to observe that the models pre-trained consistently on an Italian text corpus performed better than those pre-trained on a multi-lingual corpus, including Italian.

In general, it is observed that training that exceed the fourth epoch often result in a degradation of performance in terms of F1 score, and the ideal batch size was found to be 16. On average, the performance of all models benefits from the preprocessing step, as this strategy likely retains more informative content and includes typical social media expressions in standard Italian [14]. A significant contribution can be attributed to the translation of emojis into their Italian descriptions, which can be discriminative in emotion recognition. As a resource we used the one in [15] whose *CLDR Short Name* was translated into Italian.

The best trials were achieved with the BERT AlBERTo model, with learning rates ranging from 2e-05 to 3e-05, a patience value of 3 allowing for training for 4 epochs, a batch size of 16, and employing the "transform" preprocessing strategy. The best trial, in particular, was achieved with the following hyperparameters:

- Learning rate: 2.6263880993374053e-05
- Epsilon: 1.7454440554963499e-7
- Hidden dropout probability: 0.2
- Patience for early stopping: 3
- Batch size:

AlBERTo and dbmdz had a similar performance, however, we selected AlBERTo with an average F1 score of 0.562 also because it had a better performance in classifying fear, which was one of the most problematic due to the limited number of examples (see Figure 3).

2.5. Prediction

During the prediction phase on the test dataset made using the model fine-tuned from AlBERTo, we noticed a high number of neutral examples, and in some cases, the model was unable to determine the emotion, leaving the result field empty. We noticed this problem only on the test set and not on the validation set, therefore, to address this issue, in addition to using our trained model, we integrated the ChatGPT APIs to obtain additional results to fill in the case of neutral or undetermined sentences. Then, each sentence in the test set is pre-processed and given as input into the fine-tuned model, with a threshold set at 0.5. At the end of the prediction phase, every sentence that didn't receive any label or was classified as neutral is passed to a Python program that utilizes the ChatGPT APIs 3.5. Below, we show the prompt used and some of the examples provided to ChatGPT. Due to the limitations of the free APIs, the number of tokens and the amount of input examples are limited.

```
prompt = "Your are an emotion recognition tool for
↪  tweets and your task is to " \
  "analyze them and give a single emotion or a list of
  ↪  emotions, separated by comma that you might think
  ↪  are expressed in the current tweet and you should
  ↪  use only the emotions from " \
  "this list ['anger', 'anticipation', 'disgust', 'fear',
  ↪  'joy', 'love', 'neutral', 'sadness', 'surprise',
  ↪  'trust']"

  examples = """ \
  Here some examples:
  Input "Io ancora non ho capito se la voce mentre
  ↪  cantano sia modificata o meno
  ↪  #IlCantanteMascherato" Output:neutral
  Input "RT @user: Tartaruga is the new zoccola enorme
  ↪  #chilhavisto" Output:disgust
```

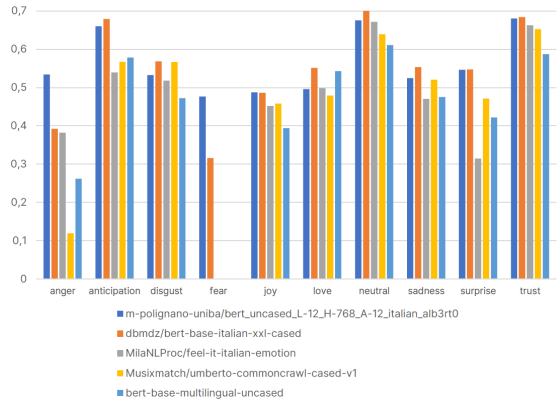

Figure 3: Comparison of the models performance.

Table 1
F1-score evaluation for in-domain testing

team	run id	anger	anticipation	disgust	fear	joy	love	neutral	sadness	surprise	trust	macro-avg
extremITA	2	0,5176	0,6420	0,6278	0,5833	0,6178	0,5190	0,7035	0,6258	0,5059	0,6854	0,6028
extremITA	1	0,4815	0,5594	0,5731	0,1429	0,5909	0,4503	0,6565	0,5233	0,4198	0,6884	0,5086
ABCD	1	0,4706	0,5946	0,5524	0,0000	0,6429	0,4586	0,6462	0,5963	0,3810	0,6516	0,4994
Emotion Hunters	1	0,4596	0,5205	0,5842	0,2400	0,4589	0,5000	0,4319	0,5484	0,4601	0,6319	0,4835
App2Check	2	0,4048	0,3814	0,5831	0,2642	0,3614	0,5463	0,3465	0,5181	0,1250	0,2108	0,3741
App2Check	1	0,3529	0,4149	0,3855	0,4000	0,6122	0,4867	0,5340	0,4339	0,3293	0,5741	0,4523
baseline_TFIDF		0,2945	0,4444	0,4680	0,3684	0,3493	0,3314	0,5392	0,3360	0,3486	0,5944	0,4074
baseline_OHE		0,2178	0,4221	0,3526	0,2593	0,2918	0,3032	0,4564	0,3191	0,2158	0,5243	0,3362
baseline_random		0,1039	0,1541	0,2683	0,0304	0,1760	0,1529	0,2941	0,1565	0,2013	0,3426	0,1872

```
Input "NE VOGLIO ANCORA #AchilleLauroExpress
↪  #pechinoexpress https://t.co/p9O3ge86i4"
↪  Output:love,trust
Input "RT @user: #UnPassoDalCielo5 stasera x il gran
↪  finale di stagione mi sono munita di Nutella
↪  biscuits" Output:anticipation,love,trust
Input "Mi manca Nicole #IlCollegio" Output:sadness
Input "Sono stata alla Risiera di San Sabba... ho
↪  ancora i brividi al solo pensiero.. #Ulisse"
↪  Output:fear
```

3. Results

The results in the following Tables 1 and 2 suggest that our approach, even if it is not the best model of the challenge, at least generalizes quite well to a domain different from the training one. This is for us a good result since we aim at applying the model in contexts different from social media analysis. In particular, we are working on the multimodal analysis of human communication with conversational agents, in which the analysis of the textual part of verbal communication can be very important to fully understand the emotional state of the user. We are actually exploring the performances of models based on LLMs by fine-tuning the most popular one on a dataset of text denoting emotion expression not taken from tweets, which is more in line with our final goal.

References

[1] O. Araque, S. Frenda, R. Sprugnoli, D. Nozza, V. Patti, EMit at EVALITA 2023: Overview of the Categorical Emotion Detection in Italian Social Media Task, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR.org, Parma, Italy, 2023.

[2] M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi, Evalita 2023: Overview of the 8th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR.org, Parma, Italy, 2023.

[3] R. Plutchik, The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice, American Scientist 89 (2001) 344–350.

[4] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, V. Basile, Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets, in: Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy,

Table 2
F1-score evaluation for out-domain testing

team	run id	anger	anticipation	disgust	fear	joy	love	neutral	sadness	surprise	trust	macro-avg
extremITA	2	0,4051	0,4923	0,6684	0,0000	0,4416	0,7552	0,6355	0,3049	0,4138	0,8603	0,4977
extremITA	1	0,5027	0,3667	0,6219	0,0000	0,3176	0,7273	0,5634	0,2024	0,3350	0,8545	0,4491
Emotion Hunters	1	0,3671	0,6053	0,6364	0,0000	0,3768	0,5907	0,6250	0,3103	0,3692	0,8632	0,4744
App2Check	2	0,6379	0,3256	0,6790	0,1818	0,2545	0,6381	0,2564	0,3195	0,1373	0,3001	0,3730
App2Check	1	0,2710	0,4301	0,4691	0,0000	0,4167	0,6528	0,3448	0,2653	0,3662	0,8064	0,4022
baseline_TFIDF		0,3972	0,4533	0,4197	0,0000	0,1890	0,3218	0,1974	0,3129	0,2812	0,7468	0,3319
baseline_OHE		0,3342	0,4412	0,4092	0,0000	0,1786	0,3034	0,1778	0,2649	0,1913	0,6869	0,2987
baseline_random		0,1984	0,0917	0,2188	0,0081	0,1624	0,2208	0,0841	0,2097	0,1336	0,5483	0,1876

November 13-15, 2019, 2019.

[5] Openai - gpt 3.5 api large language model, 2023. URL: https://chat.openai.com/chat.

[6] R. Sprugnoli, Multiemotions-it: a new dataset for opinion polarity and emotion analysis for italian, in: Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020, Bologna, Italy, March 1-3, 2021, 2020.

[7] L. C. Passaro, A. Lenci, Evaluating context selection strategies to build emotive vector space models, in: N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016, European Language Resources Association (ELRA), 2016.

[8] M. Pota, M. Ventura, R. Catelli, M. Esposito, An effective bert-based pipeline for twitter sentiment analysis: A case study in italian, Sensors 21 (2021) 133. doi:10.3390/s21010133.

[9] Mdz digital library team, «bert xxl italian models», hugging face, 2020. URL: https://huggingface.co/dbmdz/bert-base-italian-xxl-cased.

[10] L. Parisi, S. Francia, P. Magnani, Umberto: an italian language model trained with whole word masking, Original-date 55 (2020) 31Z.

[11] F. Bianchi, D. Nozza, D. Hovy, et al., Feel-it: Emotion and sentiment classification for the italian language, in: Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics, 2021.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[13] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, S. Ravi, Goemotions: A dataset of fine-grained emotions, arXiv preprint arXiv:2005.00547 (2020).

[14] M. Pota, M. Ventura, R. Catelli, M. Esposito, An effective bert-based pipeline for twitter sentiment analysis: A case study in italian, Sensors 21 (2020) 133.

[15] Emoji list, last consulted May 2023. URL: https://unicode.org/emoji/charts/full-emoji-list.html.