Original software publication

Emotion-Core: An Open Source framework for emotion detection research

Nurudin Alvarez-Gonzalez [a,*], Andreas Kaltenbrunner [a,b], Vicenç Gómez [a]

[a] *Universitat Pompeu Fabra, Barcelona, Spain*
[b] *ISI Foundation, Turin, Italy*

A R T I C L E   I N F O

Keywords:
Natural language processing
Emotion detection
Multilabel classification
Research showcase

A B S T R A C T

Identifying emotions from text is crucial for a variety of real world tasks. We describe *Emotion-Core*, an Open-Source framework for training, evaluating, and showcasing textual Emotion Detection models. Our framework is composed of two components: *Emotion Classification* and *EmotionUI*, which allow researchers to easily extend and reuse existing emotion detection solutions. We discuss the potential impact of our software project, including a recent publication in the findings of the International conference on Empirical Methods in Natural Language Processing (EMNLP 2021). Our code is available and free to use for interested researchers.

Code metadata

Current code version	v1.0.1
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-161
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/8318602/tree/v1
Legal Code License	MIT.
Code versioning system used	git
Software code languages, tools, and services used	Python, Bash, Javascript, Vue.js, AWS S3 (public deployment)
Compilation requirements, operating environments & dependencies	Managed through requirements.txt (*Emotion Classification*) and package.json (*EmotionUI*).
If available Link to developer documentation/manual	Not available.
Support email for questions	nuralgon@gmail.com

1. High-level functionality and purpose

Given a text snippet, what are the possible emotions present in it, and how strongly are they expressed? Currently, there is no Open-Source framework to answer these questions, providing pre-trained models using large-scale data sets while being easily usable and extensible. In this contribution, we provide *Emotion-Core*, a collection of Open-Source tools to train, evaluate, deploy and showcase emotion detection models using a variety of datasets, modeling techniques, and evaluation approaches. It is composed of two separate sub-projects: *Emotion Classification* and *EmotionUI*, shown in Fig. 1.

1.1. Emotion classification: Modeling and deployment back-end

Emotion Classification is a Python project containing software tools to train, evaluate and deploy machine learning models for emotion detection. The functionality is split among two main programs:

- run.py: a single-argument program that executes an experiment configuration (declaring a deterministic 'run') from a JSON file, containing all the parameters to define how to perform model-training, evaluation and serialization.
- serve.py: a web-server that exposes release-ready research models through a REST API, so that emotion detection models can be explored using a simplified web interface such as *EmotionUI*.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: nuralgon@gmail.com (N. Alvarez-Gonzalez), kaltenbrunner@gmail.com (A. Kaltenbrunner), vicen.gomez@upf.edu (V. Gómez).
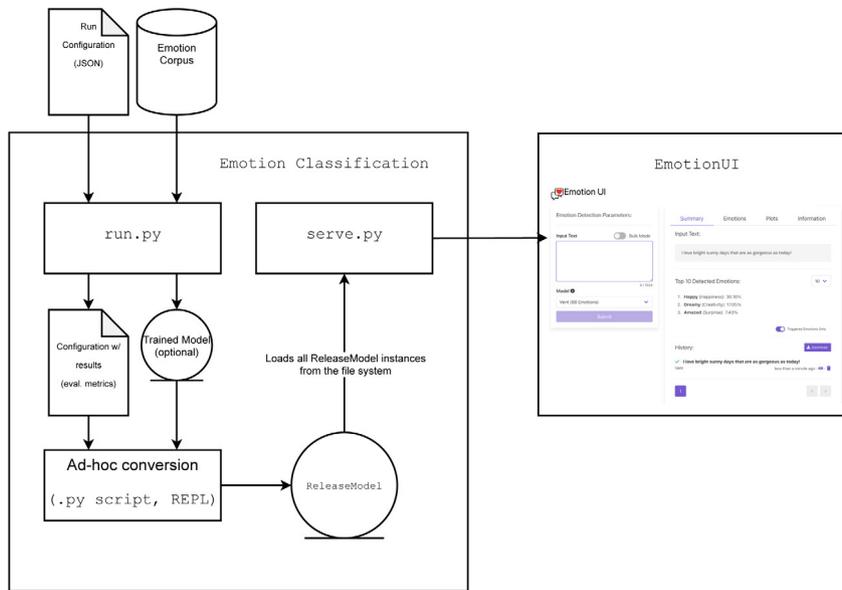
Fig. 1. *Emotion-Core* component flow chart. The diagram shows the inputs and outputs of the various systems in *EmotionUI*, from the run configuration and input data in `run.py`, the internal processing and output metrics, the preparation of a `ReleaseModel` instance for `serve.py`, and the common usage of *EmotionUI*.

Run configurations determine how to produce and evaluate an emotion detection model, with four categories of parameters:

1. **Data Loading.** Initial random seeds, split sizes for training, validation and test sets, or the name of a data column within the data set defining the different splits.
2. **Text Representations.** Define how to encode text — currently supports statistical document representations (Bag-of-Words, TF-IDF), word embeddings (FastText), or Transformer LMs (BERT).
3. **Modeling Approaches.** Declare the (streaming) modeling algorithm and hyper-parameters, with dependencies on the text representation. Includes support for traditional machine learning methods (Logistic Regression, Naïve Bayes, Random Forests) and Deep Learning approaches (Deep Neural Networks and Bi-LSTMs).
4. **Outputs and Serialization.** Specify file system paths to store model evaluation results, tuned classifier thresholds, and the trained model for deployment.

1.2. EmotionUI: Showcase and visualization interface

EmotionUI is a JavaScript project containing a single-page web application to showcase and explore emotion detection models trained with the *Emotion Classification* programs. It provides a user interface to submit text snippets both individually and in bulk, with tools to visualize the distribution of predicted emotions and emotion categories if available. The public interface is shown in Fig. 2.

2. Research impact

Emotion-Core allows researchers to rapidly experiment with emotion detection data sets, document representations, and modeling approaches in a fully reproducible way. The platform was designed as a means to benchmark emotion detection models on two of the largest emotion classification corpora: GoEmotions [2] and Vent [3]. Currently, *Emotion-Core* only supports streaming multi-label modeling approaches, with an automatic evaluation module to ensure fair comparison across all methods. Within the *Emotion-Core* framework, modelers do not need to reimplement models for each data set. Instead, the experimental methods can be extended by providing data or new implementations matching simple interfaces:

- **Data formats.** Emotion data sets in *Emotion-Core* only require two columns: `text` and `emotion_index`. An experimenter can use a new data set by preprocessing the columns accordingly. Minimal support for managing data set splitting, as well as using already provided splits, is also provided.
- **Document Representation and Models.** Representation and model factories allows for centralized definitions of how to represent a document and train a model from it. Experimenters can redefine existing methods or use new ones by creating new `Extractor` and `Model` classes and registering them in their corresponding factories.

3. Limitations, extended applications and future work

Emotion Classification currently provides users with ready-made recipes to train, evaluate and serialize models. However, deploying models is currently a manual endeavor, requiring the user to instantiate and serialize a `ReleaseModel` instance through an ad-hoc process (e.g. using an interactive Python REPL or a custom script). `ReleaseModel` instances can then be serialized to disk and automatically served by `serve.py` without additional work. A future release is intended to add automated `ReleaseModel` support, saving users from doing manual exports combining the serialized results dictionary with the serialized model.

EmotionUI provides a web interface to evaluate and explore `ReleaseModel` instances served through the `serve.py` REST API. Currently, the host name of the REST API is not configurable in real-time, being set to either a local endpoint during debugging mode or a remote production environment in the public release. This limits the ease for experimenters to evaluate multiple versions at once using different environments. A future release will add support to configure a pool of API endpoints to allow experimenters to test alternative `ReleaseModel` instances from within the web interface.

4. Related publications using *Emotion-Core*

Emotion-Core was the platform used to benchmark emotion detection models in [4], which analyzed the limits of emotion detection on two of the largest available emotion detection corpora from the perspectives of readers and writers.
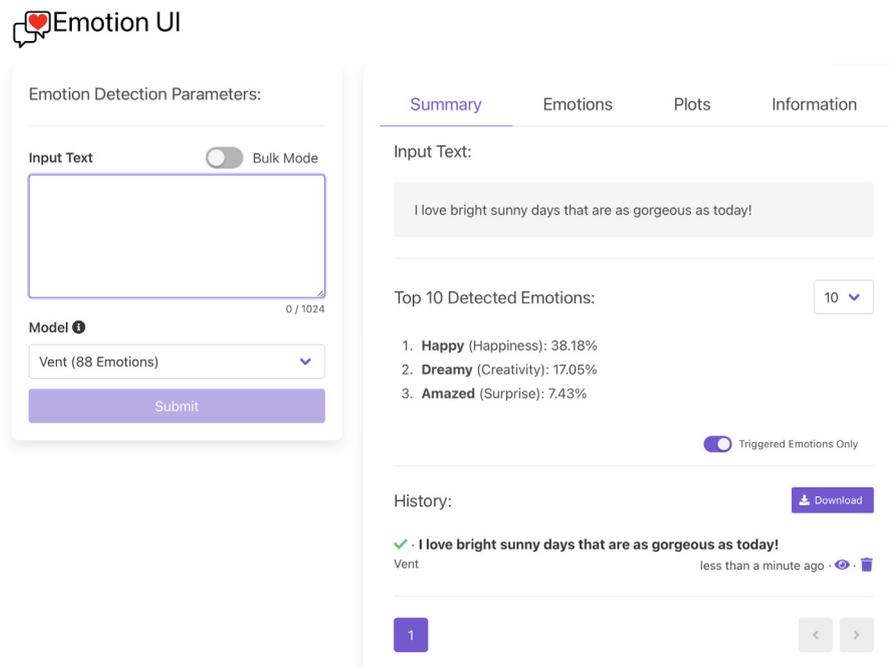
Fig. 2. Screenshot of the public *EmotionUI* [1]. Users can submit text to be classified using a model trained on different data sets, using a variety of document representations and modeling choices. The public release only exposes the best performing models on each data set.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

[1] Nurudin Alvarez-Gonzalez, Andreas Kaltenbrunner, Vicenç Gómez, EmotionUI: A web interface to explore emotion detection models, 2021, URL http://emotionui.nur.systems/, Public Release at http://emotionui.nur.systems/. Last accessed on 8th November, 2021.

[2] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, Sujith Ravi, Goemotions: A dataset of fine-grained emotions, in: 58th Annual Meeting of the Association for Computational Linguistics, 2020.

[3] Nikolaos Lykousas, Constantinos Patsakis, Andreas Kaltenbrunner, Vicenç Gómez, Sharing emotions at scale: The vent dataset, in: 13th International AAAI Conference on Weblogs and Social Media, Vol. 13, (01) 2019, pp. 611–619, URL https://www.aaai.org/ojs/index.php/ICWSM/article/view/3361.

[4] Nurudin Alvarez-Gonzalez, Andreas Kaltenbrunner, Vicenç Gómez, Uncovering the limits of text-based emotion detection, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2560–2583.