



Will Affective Computing Emerge From Foundation Models and General Artificial Intelligence? A First Evaluation of ChatGPT

Mostafa M. Amin , University of Augsburg, 86159, Augsburg, Germany, and SyncPilot GmbH, 86156, Augsburg, Germany

Erik Cambria , Nanyang Technological University, 639798, Singapore

Björn W. Schuller , University of Augsburg, 86159, Augsburg, Germany, and Imperial College London, SW7 2AZ, London, U.K.

ChatGPT has shown the potential of emerging general artificial intelligence capabilities, as it has demonstrated competent performance across many natural language processing tasks. In this work, we evaluate the capabilities of ChatGPT to perform text classification on three affective computing problems, namely, big-five personality prediction, sentiment analysis, and suicide tendency detection. We utilize three baselines, a robust language model (RoBERTa-base), a legacy word model with pretrained embeddings (Word2Vec), and a simple bag-of-words (BoW) baseline. Results show that the RoBERTa model trained for a specific downstream task generally has a superior performance. On the other hand, ChatGPT provides decent results and is relatively comparable to the Word2Vec and BoW baselines. ChatGPT further shows robustness against noisy data, where the Word2Vec model achieves worse results due to noise. Results indicate that ChatGPT is a good generalist model that is capable of achieving good results across various problems without any specialized training; however, it is not as good as a specialized model for a downstream task.

With the advent of increasingly large-data-trained general-purpose machine learning models, a new era of “foundation models” has started. According to Bommasani et al.,¹ these are marked by having been trained on “broad” data—often self-supervised—at scale leading to 1) homogenization (i.e., most use the same model for fine-tuning and training for downstream tasks, as they are effective across many tasks and too cost-intensive to train individually) and 2) emergence (i.e., tasks can be solved that these models were not originally trained upon—potentially even without additional fine-tuning or downstream training). However, at this time, much more research is needed to understand the actual emergence abilities

that potentially lead to a massive shift of paradigm in machine learning. Models might not need to be trained any more at all specifically for limited tasks, be it from the upstream or downstream perspective. Here, we consider the example of affective computing tasks seen from a natural language processing (NLP) end. In the future, will we need to train extra models at all to tackle tasks such as personality, sentiment, or suicidal tendency recognition from text, or will “big” foundation models suffice with their emergence of these?

To this end, we consider ChatGPT as our basis for a “big” foundation model to check for the full emergence of these three tasks. It was launched on 30 November 2022 and gained more than 1 million users within one week.² It has shown very promising results as an interactive chatting bot that is capable, to a large extent, of understanding questions posed by humans and giving meaningful answers to them. ChatGPT is one of the

named “foundation models” constructed by fine-tuning a large language model—namely, GPT-3—that can generate English text. The model is fine-tuned using reinforcement learning from human feedback (RLHF),³ which makes use of reward models that rank responses based on different criteria; the reward models are then used to sample a more general space of responses.^{3,4} As a result, general artificial intelligence (AI) capabilities emerged from this training mechanism, which resulted in a very fast adoption of ChatGPT by many users in a very short time.² The effectiveness of these capabilities is not exactly known yet; for example, Borji⁵ explored many of the systematic failures of ChatGPT. Zhou et al.⁶ explain the history of the development of the NLP literature until arriving at the point when ChatGPT was developed.

In summary, the aim of this article is to systematize an evaluation framework for evaluating the performance of ChatGPT on various classification tasks to answer the question of whether it shows full emergence features of other (affective computing-related) NLP tasks. We use this framework to show if ChatGPT has general capabilities that could yield competent performance on affective computing problems. The evaluation compares it against specialized models that are specifically trained on the downstream tasks. The contributions of this article are as follows:

- ▶ We evaluate whether NLP foundation models can lead to “full” (i.e., no need for fine-tuning or downstream training) emergence of other tasks, which would usually be trained on specific data sources.
- ▶ Therefore, we introduce a method to evaluate ChatGPT on classification tasks.
- ▶ We compare the results of ChatGPT on three classification problems in the field of affective computing. The problems are big-five personality prediction, sentiment analysis, and suicide and depression detection.

The remainder of this article is organized as follows. We begin by elaborating on the related work, then we introduce our method, and then we present and discuss the results. We finish with concluding remarks.

RELATED WORK

We focus on related work within the key research question of potential emergence (in the text domain) by foundation models. In particular, Qin et al.⁷ explore the question of whether ChatGPT is a general NLP solver that works for all problems. They explore a wide range of tasks, like reasoning, text summarization, named entity recognition, and sentiment analysis.

Hendy et al.⁸ explore the capabilities of GPT language models (including ChatGPT) in machine translation. Borji⁵ explores the systematic errors of ChatGPT.

METHODS

The aim of this article is to evaluate the generalization capabilities of ChatGPT across a wide range of affective computing tasks. To assess this, we utilize three datasets corresponding to three different problems, as mentioned: big-five personality prediction, sentiment analysis, and suicide tendency assessment. For these tasks, we utilize three datasets. On each of the introduced tasks, we attempt to get ChatGPT’s assessment about each of the examples of the corresponding test set. Furthermore, we compare ChatGPT against three baselines, namely, a large language model, a word model with pretrained embeddings, and a basic bag-of-words (BoW) model without making use of any external data. We describe the datasets, querying procedure of ChatGPT, and baselines in this section. Figure 1 demonstrates the pipelines of all methods (ChatGPT and the three baselines).

Datasets

We introduce the three datasets in this section. A summary of their statistics is presented in Table 1. We utilize publicly available datasets for reproducibility.

Personality Dataset

We utilize the First Impressions dataset^{9,10} for the personality task.^a Personality is represented by the big-five personality traits, namely, *openness to experience*, *conscientiousness*, *extraversion*, *agreeableness*, and *neuroticism*. The dataset consists of 15-s videos with one speaker, whose personality was manually labeled. Such labeling was conducted by relative comparisons between pairs of videos by ranking which person scores higher on each one of the big-five personality traits. A statistical model was then used to reduce the labels into regression values within the range [0, 1]. The personality labels were given based on the multiple modalities of a video, namely, images, audio, and text (content). We utilize the transcriptions of these videos as the input to be used to predict personality. We use the train/development (dev)/test split given by the publishers of the dataset.^{9,10} Like Kaya et al.,¹¹ we train the models on this dataset as a regression problem (by using mean absolute error as the loss function) since the continuous ground truth values can give a more granular estimation of the labels can give a more

^aWe acquired the dataset on 3 February 2023 from <https://chalearnlap.cvc.uab.cat/dataset/24/description/>.

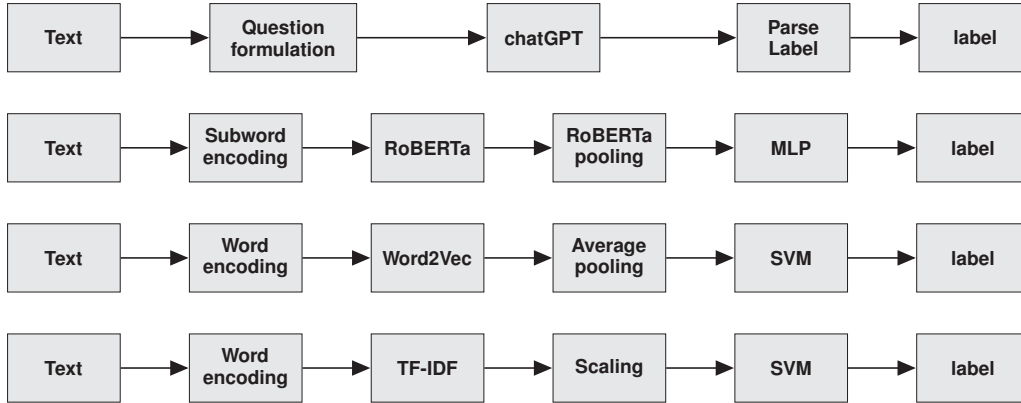


FIGURE 1. Pipelines of the ChatGPT (top row), RoBERTa baseline (second row), Word2Vec baseline (third row), and bag-of-words baseline (bottom row) approaches. MLP: multilayer perceptron; SVM: support vector machine; TF-IDF: term frequency–inverse document frequency.

TABLE 1. Statistics on the sizes of the datasets, with counts of positive and negative classes in the test set.

Dataset	Train	Dev	Test	Positive	Negative
O	6000	2000	509	333	176
C				286	223
E				214	295
A				340	169
N				274	235
Sent	1,440,144	159,856	359	182	177
Sui	138,479	6270	496	165	331

The “Test” column shows the final number of samples used for evaluation (lower than the original sizes due to the limitation of manually collecting examples from ChatGPT). A: agreeableness; C: conscientiousness; E: extraversion; N: neuroticism; O: openness to experience; Sent: sentiment; Sui: suicide.

granular estimation of the labels; then, we binarize these to positive or negative using the threshold 0.5.

Sentiment Dataset

We adopt the Sentiment140 dataset¹² for the sentiment analysis task.^b The dataset is collected from tweets on Twitter, which makes the text very noisy, which can pose a challenge for many models (especially word models). The dataset consists of tweets and the corresponding sentiment labels (positive or negative). We split the training portion with a ratio of 9:1 to give the train and dev portions^c listed in Table 1. The test portion consists of 497 tweets only; however, these were filtered down to 359 because the remaining have a *neutral* label, which is not present in the training set.

^bWe acquired the dataset from <https://huggingface.co/datasets/sentiment140> on 9 February 2023.

^c[Online]. Available at: <https://github.com/senticnet/chatgpt-affect>.

Suicide and Depression Dataset

The suicide and depression dataset¹³ is collected from the Reddit platform, under different subreddit categories, namely, “SuicideWatch,” “depression,” and “teenagers.”^d The texts of the posts from the “teenagers” category are labeled as negative, while the texts from the other two categories are labeled as positive. We excluded examples longer than 512 characters and then divided the dataset into three portions: train, dev, and test.

ChatGPT Querying Mechanism

We introduce the stages of querying ChatGPT as shown in Figure 1. The general mechanism to collect for our experiments is achieved by the following procedure for each problem:

^dWe acquired the dataset on 28 January 2023 from <https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch>.

- 1) Reformat all of the texts of the test portion of the dataset by using a format that asks ChatGPT what its guess is about the label of the text.
- 2) Chunk the examples into 25 examples per chunk.
- 3) For each chunk, open a new ChatGPT conversation.
- 4) Ask ChatGPT (manually) the reformatted question for each example, one by one, and collect the answers.
- 5) Repeat the steps 3–4 until the predictions for the whole test set are finished.
- 6) Postprocess the results in case they need some cleanup.

The formatting in the first step and the postprocessing in the last step are specified in the following two sections. We used the version of ChatGPT released on 30 January 2023.⁶

Question Formulation

The formats that are used for the three problems are given by the following snippets. The example text is substituted in place of the `{text}` part; however, the quotation marks are kept since it specifies for ChatGPT that this is a placeholder used by the question being asked. The formulations for the three problems are given as follows:

- › For the big-five personality traits, we formulate the following question: “What is your guess for the big-five personality traits of someone who said `{text}`? Answer ‘low’ or ‘high’ with bullet points for the five traits. It does not have to be fully correct. You do not need to explain the traits. Do not show any warning after.”
- › For the sentiment analysis, we formulate the following question: “What is your guess for the sentiment of the text `{text}`? Answer ‘positive,’ ‘neutral,’ or ‘negative.’ It does not have to be correct. Do not show any warning after.”
- › For the suicide problem, we formulate the following question: “What is your guess as to whether a person saying `{text}` has a suicide tendency or not? Answer ‘yes’ or ‘no.’ It does not have to be correct. Do not show any warning after.”

The formulation of the question is of crucial importance to the answer ChatGPT will give; we encountered the following aspects:

- › Asking the question directly without asking about a guess made ChatGPT, in many instances, answer

that there is little information provided to answer the question and that it cannot answer it exactly. Hence, we ask it to guess the answer, and we declare that it is acceptable to be not fully accurate.

- › It is important to ask *what* the guess is and not *Can you guess?* because this can give a response similar to the previous point, where ChatGPT responds with an answer that starts with *No, I cannot accurately answer whether. . .* Therefore, the question needs to be assertive and specific.
- › We need to specify the exact output format, because ChatGPT can get innovative otherwise about the formatting of the answer, which can make it hard to collect the answers for our experiment. Despite specifying the format, it still sometimes gave different formats. We elaborate on this in the next section.
- › The questions for the suicide assessment task triggered warnings in the responses of ChatGPT due to their sensitive content. We elaborate on the terms of use in the “Acknowledgments” section.

Parsing Responses

The responses of ChatGPT need to be parsed since ChatGPT can give arbitrary formats for a given answer, even when the content is the same. This is predominant in the personality traits since there are five traits. Sometimes, the answers are listed as bullet points; other times, they are all in one comma-separated line.

Also, it uses different delimiters or order, e.g., “Openness: Low,” “Low in Openness,” and “Low: Openness.” Additionally, in all problems, it sometimes gives an introduction for the answer, for example, “Here is my guess for . . .” or “Based on the statement. . .” We counter this issue by using regular expressions to capture the responses.

Baselines

To compare the performance of ChatGPT on the different tasks, we need to use baselines and train them on the train portion (while validating on the dev portion). We employ three baselines, which serve as the specialized models specifically tailored for the corresponding downstream task. The first baseline is a robust language model (RoBERTa) trained on a large amount of text. The second is a simple baseline that uses a word model by employing pretrained Word2Vec embeddings on the words of a sentence with a simple classifier. The third baseline is a simple BoW model that utilizes a linear classifier. The hyperparameters of all models are optimized by selecting the hyperparameters yielding

⁶ChatGPT release notes: <https://help.openai.com/en/articles/6825453-chatgpt-release-notes>.

TABLE 2. Hyperparameters of the different baselines.

Target Label	RoBERTa			W2V	BoW
	N	U	α	C	η
O	2	498	5.66×10^{-4}	0.0378	2.47×10^{-3}
C				0.0472	3.09×10^{-6}
E				0.0069	1.09×10^{-5}
A				0.0218	4.65×10^{-4}
N				0.0657	2.21×10^{-6}
Sen	3	420	2.97×10^{-5}	0.0144	5.25×10^{-6}
Sui	3	497	8.04×10^{-4}	10.00	4.71×10^{-6}

N is the number of hidden layers in the multilayer perceptron using RoBERTa representations, U is the number of neurons in the first hidden layer (which is halved for each subsequent layer), and α is the learning rate. The Adam optimizer always yields the best results as compared to stochastic gradient descent (SGD). C is the support vector machine parameter for Word2Vec, and the sentiment model used linear kernel, while the other models used the radial basis function kernel. η is the learning rate of the SGD in the BoW model. Bow: bag of words; W2V: Word2Vec.

the best performance on the dev portion. The hyperparameters are tuned using the SMAC toolkit,¹⁴ which is based on Bayesian optimization. The selected hyperparameters are listed in Table 2.

RoBERTa Language Model

The baseline RoBERTa¹⁵ is a pretrained BERT model, which has a transformer architecture. Liu et al.¹⁵ trained two instances of RoBERTa; we use the smaller one, namely, *RoBERTa-base*,^f consisting of 110 million parameters. RoBERTa-base is pretrained on a mixture of several large datasets that included books, English Wikipedia, English news, Reddit posts, and stories.¹⁵ The model starts by tokenizing a text using subword encoding, which is a hybrid representation between character-based and word-based encodings. The tokens are then fed to RoBERTa to obtain a sequence of embeddings. The pooling layer of RoBERTa is then used to reduce the embeddings into one embedding only, hence acquiring a static feature vector of size 768 representing the text. We additionally train a multilayer perceptron (MLP)¹⁶ to predict the final label. The pipeline for the model is shown in Figure 1.

For the training procedure, we use SMAC¹⁴ to select the MLP specifications. We employ SMAC to sample a total of 100 models per task and train them with a batch size 256 for 300 epochs with early stopping to prune the ineffective models. Eventually, the model with the best performance on the dev set is selected. The hyperparameter space consists of four hyperparameters: the number of hidden layers $N \in [0, 3]$, the number of neurons in the first hidden layer $U \in [64, 512]$ (log sampled),

the optimization algorithm [Adam¹⁷ or stochastic gradient descent (SGD)¹⁶], and the learning rate $\alpha \in [10^{-6}, 10]$ (log sampled). The number of neurons in the hidden layers is specified by the first one as a hyperparameter; then, the number of neurons is halved for each subsequent hidden layer (clipped to be at least 32). The hidden layers have rectified linear unit as an activation function. The final layer has a sigmoid activation function. The loss function for classification is cross entropy and for regression is mean absolute error.

Word2Vec Word Embeddings

The baseline Word2Vec^{18,19} makes use of pretrained word embeddings,^g which are trained on a large amount of text from Google News. The model operates by tokenizing a given text into words; each word is assigned an embedding from the pretrained embeddings. The embeddings are then averaged for all words to give a static feature vector of size 300 for the entire string. A support vector machine (SVM)¹⁶ is then used to predict the given task. The pipeline of this model is shown in Figure 1.

We train the SVM model by tuning its hyperparameter C using SMAC¹⁴ by sampling 20 values within the range $[10^{-6}, 10^4]$ (log sampled) and choosing the model that yields the best score on the dev set. We use the radial basis function (RBF) kernel for the SVMs except for the sentiment dataset, where we apply a linear kernel, as the sentiment dataset is much bigger (as shown in Table 1), which renders the RBF impractical due to the computational efficiency.

^fAcquired on 9 February 2023 from <https://huggingface.co/docs/transformers/modeldoc/roberta>.

^gAcquired on 16 February 2023 from <https://code.google.com/archive/p/word2vec/>.

BoW

The BoW model is a very simple baseline that does not rely on any knowledge transfer or large-scale training. In particular, it uses only in-domain data for training and no other data for either up- or downstreaming.

We utilize the classical technique term frequency-inverse document frequency, which tokenizes the sentences into words; then, a sentence is represented by a vector of the counts of the words it contains. The vector is then normalized by the term frequency across the entire train set of the corresponding dataset. We restrict the words to the most common 10,000 words in the train set; then, we scale each feature to be within $[-1, 1]$, by dividing by the maximum absolute value of the feature across the train set. We optimize a linear kernel SVM, and we optimize using SGD¹⁶ due to the high number of features (10,000 features). We tune the learning rate η of SGD using SMAC.¹⁴

RESULTS

In this section, we review the results of our experiments. In summary, we evaluate the performance of ChatGPT against the three baselines—RoBERTa, Word2Vec, and BoW—on three downstream classification tasks, namely, personality traits, sentiment analysis, and suicide tendency assessment. We measure classification accuracy and unweighted average recall (UAR)²⁰ as performance measures. UAR has the advantage of exposing whether a model is performing very well on a class at the expense of the other class, especially in imbalanced datasets. Additionally, we utilize the randomized permutation test as a statistical significance test.²¹ The main results of the experiments are shown in Table 3.

DISCUSSION

The RoBERTa model is achieving the best performance for the personality and suicide assessment tasks, with a statistically significant improvement of accuracy over ChatGPT. However, ChatGPT is the best in sentiment analysis, but only slightly better than RoBERTa. The UAR for the personality traits points to similar conclusions about the relative performance; however, it yields much lower values for all baselines on some of the traits (openness and agreeableness). The UAR measure generally yields similar results on all models for both sentiment analysis and suicide assessment. The performance of ChatGPT on the personality assessment is inferior to that of the three baselines on all of the traits. It is significantly worse than RoBERTa on all traits and Word2Vec on three traits.

ChatGPT has the best performance in the sentiment analysis, where it is slightly better than RoBERTa and BoW and significantly better than Word2Vec. One of the potential reasons for the inferiority of Word2Vec and BoW on the sentiment dataset is not using subword encodings. The reason is that the sentiment dataset is collected from Twitter, so it is very noisy, which can lead to many mistakes in identifying the words and, hence, assigning them the proper embeddings. Subword encoding avoids many of these issues since a few typos would still yield a meaningful subword representation of the given sentence.

The results on the suicide assessment problem show the contrast between the aforementioned analyses. The task is not as hard as the personality assessment problem, with a much bigger amount of training data. The suicide assessment can, rather, be thought of as classifying extreme negative sentiment, where

TABLE 3. The classification accuracy and unweighted average recall of ChatGPT against the baselines on the different tasks.

Target Label	Accuracy				Unweighted Average Recall (%)			
	ChatGPT	RoBERTa	Word2Vec	BoW	ChatGPT	RoBERTa	Word2Vec	BoW
O	46.6	66.0 ***	65.2***	59.7***	50.1	50.9	50.7	55.6
C	57.4	63.7 *	62.7	55.6	57.7	60.8	60.0	56.3
E	55.2	66.0 ***	59.9	55.2	54.0	62.3 ***	55.5	53.7
A	44.8	67.4 ***	67.2***	58.5***	48.4	51.9	51.0	55.7 *
N	47.2	62.1 ***	56.8***	56.0***	49.1	61.2 ***	54.6	55.8*
Sen	85.5	85.0	79.4*	82.5	85.5	85.0	79.4**	82.4
Sui	92.7	97.4 ***	92.1	92.7	91.2	97.4 ***	91.2	90.9

*Statistically significant difference as compared to ChatGPT, with a p value of 0.05. **Statistically significant difference as compared to ChatGPT, with a p value of 0.02. ***Statistically significant difference as compared to ChatGPT, with a p value of 0.01. Significance tests are checked with a randomized permutation test. The bold values refer to the best model (out of the four presented models, namely ChatGPT and the three baselines) for a specific combination of target label and performance metric.

Qin et al.⁷ showed that ChatGPT is better at predicting negative sentiment than positive. However, the texts of the suicide dataset are much less noisy compared to those of the sentiment dataset. In that case, the performances of the Word2Vec and BoW models are more or less on par with that of the ChatGPT model, while RoBERTa is significantly better than all of them.

Our experiments indicate that ChatGPT has a decent performance across many tasks (especially sentiment analysis or similar tasks), which is comparable to simple specialized models that solve a downstream task. However, it is not competent enough as compared to the best specialized model to solve the same downstream task (e.g., fine-tuned RoBERTa). The performance of ChatGPT does not generally show statistically significant differences when compared to the simplest baseline BoW, which does not make any use of pretraining. This is further confirmed by Hendy et al.⁸ in machine translation and other tasks.⁷ In summary, our study suggests that ChatGPT is a generalist model (in contrast to a specialized model) that can decently solve many different problems without specialized training. However, to achieve the best results on specific downstream tasks, dedicated training is still required. This might be enhanced in future versions of ChatGPT and similar models by including more diverse tasks for the RLHF component in the training.

Limitations

The most crucial limitation of the presented results is the small amount of data for evaluation (497, 362, and 509 examples for the three tasks) since ChatGPT is only available for manual entries by consumers and not for automated large-scale testing. Additionally, it only responds to approximately 25–35 requests per hour to reduce the computational cost and avoid brute forcing. Another issue that may limit future experiments is parsing the responses. In our experiments, ChatGPT responded with arbitrary formatting despite the fact that we specified the desired format explicitly in the question prompt.

CONCLUSION

In this article, we provided first insight into the potential “full” emergence of tasks by broad-data-trained foundation models. We approached this from the perspective of natural language tasks in the affective computing domain and chose ChatGPT as the exemplary foundation model. To this end, we introduced a framework to evaluate the performance of ChatGPT as a generalist foundation model against specialized models on a total of seven classification tasks from three affective

computing problems, namely, personality assessment, sentiment analysis, and suicide tendency assessment.

We compared the results against three baselines, which reflect training the downstream tasks and using or not using additional data for the upstream task. The first model was RoBERTa, a large-scale-trained, transformer-based language model; the second was Word2Vec, a deep learning model trained to reconstruct the linguistic contexts of words; and the third was a simple BoW model.

The experiments have shown that ChatGPT is a generalist model that has a decent performance on a wide range of problems without specialized training. ChatGPT showed superior performance in sentiment analysis, poor performance on personality assessment, average performance on suicide assessment. In other words, we could demonstrate genuine emergence properties, potentially rendering future efforts to collect task-specific databases increasingly obsolete.

However, the performance of ChatGPT is not particularly impressive since it did not show statistically significant differences with a simple BoW model in almost all cases. On the other hand, RoBERTa, fine-tuned for a specific task, had significantly better performance as compared to ChatGPT on the given tasks, which suggests that, despite the generalization abilities of ChatGPT, specialized models are still the best option for optimal performance. However, this can be taken into consideration in future developments of foundation models like ChatGPT to yield wider exploration spaces for training.

In the near future, we will extend our experiments to more metrics, e.g., explainability and computational efficiency, on top of accuracy and UAR. We also plan to expand our comparative evaluation to more sophisticated models (e.g., prompt-based classification²² and neurosymbolic AI²³) and more advanced affective computing tasks (e.g., sarcasm detection,²⁴ metaphor understanding,²⁵ and conversational emotion recognition²⁶) but also more complex sentiment datasets requiring commonsense reasoning and/or narrative understanding.

ACKNOWLEDGMENTS

We would like to thank OpenAI for the usage of ChatGPT. We followed the policy of ChatGPT.^h Our use of ChatGPT is purely for research purposes to assess emerging capabilities of foundation models and does not promote the use of ChatGPT in any way that violates the aforementioned usage policy; in particular, with regard to the subject of self-harm, note

^hUsage policy released on 15 February 2023: <https://platform.openai.com/docs/usage-policies/disallowed-usage>.

that some of the examples in the datasets we used triggered a related warning by ChatGPT.

REFERENCES

1. R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.
2. S. Mollman. "ChatGPT gained 1 million users in under a week. Here's why the AI chatbot is primed to disrupt search as we know it." Yahoo! Finance. Accessed: Feb. 21, 2023. [Online]. Available: <https://finance.yahoo.com/news/chatgpt-gained-1-million-followers-224523258.html>
3. L. Ouyang et al., "Training language models to follow instructions with human feedback," 2022, *arXiv:2203.02155*.
4. L. Gao, J. Schulman, and J. Hilton, "Scaling laws for reward model overoptimization," 2022, *arXiv:2210.10760*.
5. A. Borji, "A categorical archive of ChatGPT failures," 2023, *arXiv:2302.03494*.
6. C. Zhou et al., "A comprehensive survey on pretrained foundation models: A history from BERT to ChatGPT," 2023, *arXiv:2302.09419*.
7. C. Qin et al., "Is ChatGPT a general-purpose natural language processing task solver?" 2023, *arXiv:2302.06476*.
8. A. Hendy et al., "How good are GPT models at machine translation? A comprehensive evaluation," 2023, *arXiv:2302.09210*.
9. V. Ponce-López et al., "Chalearn lap 2016: First round challenge on first impressions – Dataset and results," in *Proc. Eur. Conf. Comput. Vision*, Cham, Switzerland: Springer International Publishing, 2016, pp. 400–418, doi: 10.1007/978-3-319-49409-8_32.
10. H. J. Escalante et al., "Design of an explainable machine learning challenge for video interviews," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, AK, USA, 2017, pp. 3688–3695, doi: 10.1109/IJCNN.2017.7966320.
11. H. Kaya, F. Gurpinar, and A. A. Salah, "Multi-modal score fusion and decision trees for explainable automatic job candidate screening from video CVs," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR) Workshops*, Honolulu, HI, USA, 2017, pp. 1–9.
12. A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," Stanford University, Stanford, CA, USA, CS224N project report, 2009.
13. V. Desu et al., "Suicide and depression detection in social media forums," in *Proc. Smart Intell. Comput. Appl.*, Singapore: Springer Nature, 2022, vol. 2, pp. 263–270.
14. M. Lindauer et al., "SMAC3: A versatile bayesian optimization package for hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 23, no. 54, pp. 1–9, Jan. 2022.
15. Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
16. C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
17. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015, *arXiv:1412.6980*.
18. T. Mikolov et al., "Efficient estimation of word representations in vector space," 2013.
19. T. Mikolov et al., "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA: Curran Associates, Inc., 2013, pp. 3111–3119.
20. B. Schuller et al., "The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism," in *Proc. 14th Annu. Conf. Int. Speech Commun. Assoc. (ISCA)*, Lyon, France, 2013, pp. 148–152, doi: 10.21437/Interspeech.2013-56.
21. P. Good, *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. New York, NY, USA: Springer-Verlag, 1994.
22. R. Mao et al., "The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection," *IEEE Trans. Affective Comput.*, early access, 2023, doi: 10.1109/TAFFC.2022.3204972.
23. E. Cambria et al., "SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis," in *Proc. 13th Lang. Resour. Eval. Conf. (LREC)*, 2022, pp. 3829–3839.
24. N. Majumder et al., "Sentiment and sarcasm classification with multitask learning," *IEEE Intell. Syst.*, vol. 34, no. 3, pp. 38–43, Jul. 2019, doi: 10.1109/MIS.2019.2904691.
25. M. Ge, R. Mao, and E. Cambria, "Explainable metaphor identification inspired by conceptual metaphor theory," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, pp. 10,681–10,689, doi: 10.1609/aaai.v36i10.21313.
26. W. Li et al., "SKIER: A symbolic knowledge integrated model for conversational emotion recognition," in *Proc. 37th AAAI Conf. Artif. Intell.*, 2023.

MOSTAFA M. AMIN is working toward his Ph.D. with the Chair of Embedded Intelligence for Health Care and Well-being, University of Augsburg, 86159, Augsburg, Germany, and is a senior research data scientist at SyncPilot GmbH, 86156, Augsburg, Germany. His research interests include affective computing and audio and text analytics. Amin received his

M.Sc. degree in computer science from the University of Freiburg, Germany. Contact him at mostafa.mohamed@unia.de.

ERIK CAMBRIA is an associate professor at Nanyang Technological University, 639798, Singapore. His research interests include neurosymbolic artificial intelligence for explainable natural language processing in domains like sentiment analysis, dialogue systems, and financial forecasting. Cambria received his Ph.D. degree in computing science and mathematics through a joint program between the University of Stirling and the Massachusetts Institute of Technology Media Lab. He is a Fellow of IEEE. Contact him at cambria@ntu.edu.sg.

BJÖRN W. SCHULLER is a professor of artificial intelligence with the Department of Computing, Imperial College London, SW7 2AZ, London, U.K., where he heads the Group on Language, Audio, and Music. He is also a full professor and the head of the Chair of Embedded Intelligence for Health Care and Wellbeing with the University of Augsburg, 86159, Augsburg, Germany and the founding chief executive officer/chief scientific officer of audEERING. Schuller received his doctoral degree in electrical engineering and information technology from the Technical University of Munich. Contact him at schuller@IEEE.org.