Early detection of depression with linear models using hand-crafted and contextual features

Ilija Tavchioski[1,3], Blaž Škrlj[1], Senja Pollak[1,2] and Boshko Koloski[1,2]

[1] *Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[2] *International Postgraduate School Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[3] *Faculty of Computer and Information Sciences, Večna Pot 113, 1000 Ljubljana, Slovenia*

Abstract

Depression is a mental illness that affects millions of people; its early detection is of great importance for diagnosis and treatment. In this work, we describe our solution submitted to the joint task of *Early Detection of Depression* organized by CLEF, achieving 8th place out of 13 teams in terms of F1 score, however, performing the best in terms of precision. The result was obtained with one of the most computationally non-expensive approaches. Our approach focused on using linear models, such as logistic regression, that learned from different representations of the input space of documents.

Keywords

Depression detection, Document classification, Natural Language Processing, Machine Learning, Social Media

1. Introduction

This work presents our solution to the problem of early depression detection proposed as a shared task in [1]. Since the appearance of social media and their exponential growth throughout the past decade in terms of users, the social media had become an important part of our daily life and opened a new ways to express ourselves. A study [2] has shown that our behavior on social media is not very different from the one in real life, thus there is a possibility for inferring health related problems, especially mental illnesses such as depression from posts on social media. Due to this behaviour we consider utilizing machine learning methods to detect a sign of depression as early as possible by using natural language processing techniques. The remainder of this paper is structured as follows: in Section 2 we present the background and related work for detecting depression from texts using machine learning, then in Section 3 we describe the data that was given by the task's organizers, next, in Section 4 we describe our proposed methodology, in Section 7 we present our results obtained on the official test set and in Section 8 we present conclusions and further work.

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

ilijatavchioski@gmail.com (I. Tavchioski); blaz.skrlj@ijs.si (B. Škrlj); senja.pollak@ijs.si (S. Pollak); boshko.koloski@ijs.si (B. Koloski)

2. Background and Related work

The problem of this work is defined as follows: *Given a user and a list of their posts on the social platform Reddit, decide from which post onwards the user is depressed.* Our goal is to develop a method that will process sequentially each round of posts from all users and determine if a user is depressed. If some post is labeled as depressing we consider the user as depressed, the assessment is final and their additional posts are not to be taken into account. Prior to this year's task, there were several proposed solutions using machine learning and natural language processing in the past few years on similar or almost identical tasks. Analyzing and profiling users on social-media represents an active research area. Argamon et al. [3] did the pioneering work in author profiling on the level of British National Corpora and concluded that man intend to write in less formal way. Litvinova et al. [4] showed that the distinction via gender can be captured based on person's writing adjectives between male and female people. This hypothesis lead to several shared-tasks aimed at author-profiling [5, 6, 7, 8, 9, 10, 11, 12]. The shared-tasks profiled users on various parallels: gender, age, occupation, are the users potential spreaders of fake or hate news and so on. The highest-scoring approaches to these tasks included models that base on simpler linguistic features classified with either Support Vector Machine or Logistic Regression. Martinc et al [13] focused on creating TF-IDF weighted n-gram features based both on word and character n-grams classified via Support Vector Machines. Koloski et al. [14] improved the proposed approach by introducing singular-value-decomposition of the n-gram space. The proposed representation performed competitivly well in a multilignual setting, for the task of fake-news spreaders identification [15]. In the domain of depression detection, Basile et al. [16] proposed a solution to the task of depression detection with Hierarchical Attention Networks, constructed based on 20000 most-frequent words, initialized with the GloVe [17] embeddings. Campillo et al. [18] proposed a solution using TF-IDF weighted representations derived from the word features, while for classification they utilized the Support Vector Machine model. Key feature to this work was that the authors also included the position of a given post in a series of posts, hypothesizing that a post earlier in the sequence can have higher impact on the class prediction. The **BERT** based architectures are also commonly used for detection of depression. One of the solutions including large pre-trained models is the method by Castaño et al. [19] where the authors used **XLM-RoBERTa** [20] with an additional classification head. Transfer-learning recently gained traction as a popular paradigm of utilizing knowledge of the language model that was acquired by solving an unsupervised task. Spartailis et al. [21] used the aforementioned paradigm by utilizing **SBERT** [22] with a combination of feature extraction via classical machine learning.

3. Data set

The shared task consisted of two stages: a development and a test stage. In each stage we were given up to N users and their corresponding k posts, accompanied by a binary label (depressed or not-depressed). In the *development* stage, the organizers provided data for training that consisted of the training and test data from eRisk2017 edition and the test data from the eRisk2018 edition. A total of *1618* users were given. For each user up to *2000* posts were given.

The data was skewed towards non-depressed class, with more than 90% of the users labeled as non-depressed. In order to build and evaluate models internally, we split the data into a training (80% of the train data) and a development (20% of the train data) set with respect to class distributions. Table 1 shows the data distribution per each training split. For learning and modelling purposes, we first pair every given post per user with that user's corresponding binary label (depressed or not-depressed). In this manner we acquire ***843,554*** training data points and ***204,874*** testing data points.

Table 1
Data distribution. The users column indicates the number of users per split, while the writings column represents the total amount of posts per split.

	Training data		Development data		Test data	
Label	Total users	Total writings	Total users	Total writings	Total users	Total writings
Depressed	135 (10.57 %)	75976 (9.01 %)	42 (12.31 %)	17916 (8.744%)	98 (7 %)	35,332
Not depressed	1,142 (89.42 %)	767,578 (90.99 %)	299 (87.68 %)	186958 (91.25 %)	1,302 (93 %)	687,228
All	1277	843,554	341	204,874	1,400	722,560

4. Methodology

We treat this problem as a binary document classification problem. In this section we describe the chosen document representations, followed by the classifier description and finally we explain our final task modelling.

4.1. Document representation

We consider two different document representation methods: one based on Latent Semantic Analysis (see section 4.1.1) and one contextual based on the sentence transformers (see section 4.1.2). We use the implementation by Koloski et al. [23] in c19 python package [1]. For the classification model we use the Logistic Regression model by scikit-learn [24].

4.1.1. Latent Semantic Analysis

For our first representation method we consider the [23] implementation of LSA based on n-gram features recuded via **SVD** technique to create a new latent space of reduced dimensionality. The method has two hyper-parameters n - the total number of n-gram features and d the dimension of the latent space. The method first pre-processes the documents by removing the punctuation, the hashtags, the URLs and stop-words. Next, the POS-tags are extracted with the NLTK library [25]. The method constructs $\frac{n}{2}$ features on basis of TF-IDF weighted word uni-grams and bi-grams and $\frac{n}{2}$ features of TF-IDF weighted char bi-grams and tri-grams. Finally, **SVD** is applied in order to create the new latent space and simultaneously reduce the dimensionality to dimension d. We search extensively thought the parameter space $n = \{500, 1000, 1500, 2000\}$ and $d = \{64, 128, 256, 512\}$. In our method the best-performing hyper-parameters were set to $n = 1000$ and $d = 256$ respectively.

[1] https://github.com/bkolosk1/c19_rep

4.1.2. Contextual Features

For our second method we considered to use a model from sentence-transformer library *distilbert-base-nli-mean-tokens* [22] in order to map the writings to a dense vector space of 768 dimensions. Then, using the obtained vector representations we classify the writings using the aforementioned linear model.

Sentence-BERT is a **BERT** [26] based model that it is used is to derive a semantically meaningful vector representations for documents on sentence-level. It has added a pooling operation in order to aggregate and generate the representations. We considered the following variants of **SBERT** model: **distillBERT** [27], **RoBERTa** [28], **XLM-RoBERTa** [20].

5. Final Classification

We train a Logistic Regression Classifier on top of these aforementioned features with penalty C set to 1. We learned a classifier on a given representation on the training set and evaluated on the development set. For evaluation measure we used the F1-score.

5.1. User classification

We start by processing the first post, if that one is predicted as depressive by our system we return depressing automatically for every next post. If not, we return that the user is not depressed and proceed to classify the next post. Once we find a post that is depressing, we proceed to automatically reply that the user is depressed until we finish. If there is no depressing post we return that the user is not depressed at every step up to the last query to our system.

6. Measures

The F1-score is calculated as a harmonic mean between precision and recall.

- **Precision** is a metric that provides us with the percentage of the positive predictions that are actually positive in the test set.
- **Recall** is a metric that provides us with the proportion of the positive instances in the test set that are predicted as positive by our model.
- **F1-score** is a harmonic mean of the precision and recall score.
- **ERDE** [29] is a metric that, was provided by the task's organizers, and penalizes the late response for correct predictions for depression.
- **speed** is a metric that presents how fast the model predicts positive for the instances that are labeled as depressed.

7. Experiments and Evaluation Results

In the following Section we will describe the evaluation settings with the corresponding measures and the evaluation results - both on the internal and official test set.

7.1. Internal evaluation

In this subsection we explain our internal experimentation setup that was performed on the internal data split defined in Section 2, followed by the presentation of the results obtained in Table 2. As described in the previous sections, we first construct a representation (in the case of **LSA**) or obtain it directly (in the case of **sBERT**) on the training set, and next we train a classifier that we evaluate on the development set. The **LSA** representation outscored the four other models in term of precision, achieving score of *0.5385*. **DistilBERT** was next in line falling behind by *0.2301* percentage points in terms of precision, that was followed by **XLM-RoBERTa** and **RoBERTa**. In terms of recall best performing model was the **RoBERTa** model achieving score of *0.9524*, followed by **XLM-RoBERTa**, while **LSA** was at the last place in terms of recall with score of *0.1667*. The best-performing model in terms of F1-score was the **DistilBERT** model achieving score of *0.4430* percentage points, followed by **XLM-RoBERTa**, **RoBERTa** and finally **LSA**. The more granular evaluation of the **DistilBERT** model is presented in Figure 1. We considered the **LSA** model for our first submission as it had highest precision, and as for the second submission we consider the **DistilBERT** since it has produced predictions with the highest F1 score.

Table 2
Internal Evaluation Results.

Method	Precision	Recall	F_1 score
distillBERT	0.3084	0.7857	**0.4430**
RoBERTa	0.1961	**0.9524**	0.3252
XLM-RoBERTa	0.2031	0.9286	0.3333
LSA	**0.5385**	0.1667	0.2545

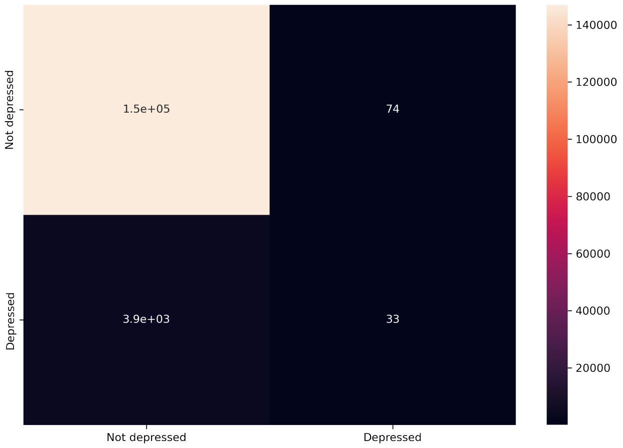

Figure 1: Confusion-matrix of the predictions of the **DistilBERT** model on the internal test set.

7.2. Official Test Set Results

Table 3 represents the results achieved on the official test set provided by the organizers. In addition to our results, we added the top three results performed by three other teams that also processed all 2000 writings for comparison. Our **LSA** solution achieved the best score in terms of Precision with a highest score of *0.684*, while our second run based on **RoBERTa** achieved Recall score of *0.959* falling behind only by 4.1% behind the best recall score. In terms of *latencyTP* and *speed* we achieved the best score. Finally we ranked $8th$ out of 13 places.

Table 3
Official Test Results

Method	Precision	Recall	F1	*ERDE*^{50}	*speed*	writings processed
LSA	**0.684**	0.133	0.222	0.061	**1.000**	2000/2000
CF distillBERT	0.242	**0.959**	0.387	0.036	0.924	2000/2000
SCIR2-run3	0.316	0.847	0.460	**0.026**	0.834	2000/2000
UNSL-run2	0.400	0.755	0.523	0.026	0.992	2000/2000
BLUE-run0	0.395	0.898	**0.548**	0.027	0.984	2000/2000

8. Conclusion and further work

In our attempt to solve this joint task proposed by CLEF, we considered using light machine learning models such as logistic regression on different input representations based on *LSA* and contextual features. Although we achieved a decent performance in terms of F1 score of **0.387**, we still lag behind the top results in this task. On the other hand, we achieved the best precision with the first method and almost perfect recall with the second method, which shows the performance of our method in detecting depressed users and their early detection, as this method also performed quite well on the *ERDE*50 metric. The results of the proposed method on the official results indicate that false negatives are not well captured by the model (low recall), however, false positives are easily identified – this is not necessarily optimal for a practical application, where either F1 or recall can have a greater practical relevance, however, could indicate the method's usefulness in particular scenarios aimed to identify existing patients that were mis-treated. And it is still worth mentioning that we were one of the 7 teams that processed all the fonts, thanks to the low time consumption of our model, which also leads us to achieve the best speed on the track. Of course, we make predictions based on only one writing, and this output shows our direction for further improving these methods by using multiple writings to get a more meaningful feeling from users' writings. Additional further work can be done by improving the performance of our system via making ensembles of classifiers as trying to include background knowledge or test AutoML systems for automatic feature creation and classifier selection.

9. Availability

The code can be found here: https://gitlab.com/teletton/erisk-task2-depression.

10. Acknowledgments

This work was supported by the Slovenian Research Agency (ARRS) grants for the core programme Knowledge technologies (P2-0103), the project Computer-assisted multilingual news discourse analysis with contextual embeddings (CANDAS, J6-2581).

References

[1] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Overview of erisk 2021: Early risk prediction on the internet, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2021, p. 324–344. URL: https://doi.org/10.1007/978-3-030-85251-1_22. doi:10.1007/978-3-030-85251-1_22.

[2] T. C. Marriott, T. Buchanan, The true self online: Personality correlates of preference for self-expression online, and observer ratings of personality online and offline, Comput. Hum. Behav. 32 (2014) 171–177.

[3] S. Argamon, M. Koppel, J. Fine, A. R. Shimoni, Gender, genre, and writing style in formal written texts, Text & talk 23 (2003) 321–346.

[4] T. Litvinova, O. Zagorovskaya, O. Litvinova, P. Seredin, Profiling a set of personality traits of a text's author: a corpus-based approach, in: International Conference on Speech and Computer, Springer, 2016, pp. 555–562.

[5] F. Rangel, P. Rosso, M. Koppel, E. Stamatatos, G. Inches, Overview of the author profiling task at pan 2013, in: CLEF Conference on Multilingual and Multimodal Information Access Evaluation, CELCT, 2013, pp. 352–365.

[6] F. Rangel, P. Rosso, I. Chugur, M. Potthast, M. Trenkmann, B. Stein, B. Verhoeven, W. Daelemans, Overview of the 2nd author profiling task at pan 2014, in: CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014, 2014, pp. 1–30.

[7] F. M. Rangel Pardo, F. Celli, P. Rosso, M. Potthast, B. Stein, W. Daelemans, Overview of the 3rd author profiling task at pan 2015, in: CLEF 2015 Evaluation Labs and Workshop Working Notes Papers, 2015, pp. 1–8.

[8] F. Rangel, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast, B. Stein, Overview of the 4th author profiling task at pan 2016: cross-genre evaluations, in: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings/Balog, Krisztian [edit.]; et al., 2016, pp. 750–784.

[9] F. Rangel, P. Rosso, M. Potthast, B. Stein, Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter, Working notes papers of the CLEF (2017) 1613–0073.

[10] F. Rangel, P. Rosso, M. Montes-y Gómez, M. Potthast, B. Stein, Overview of the 6th author profiling task at pan 2018: multimodal gender identification in twitter, Working Notes Papers of the CLEF (2018) 1–38.

[11] F. Rangel, P. Rosso, Overview of the 7th author profiling task at pan 2019: bots and gender

profiling in twitter, in: Working Notes Papers of the CLEF 2019 Evaluation Labs Volume 2380 of CEUR Workshop, 2019.

[12] F. Rangel, A. Giachanou, B. H. H. Ghanem, P. Rosso, Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter, in: CEUR Workshop Proceedings, volume 2696, Sun SITE Central Europe, 2020, pp. 1–18.

[13] M. Martinc, I. Skrjanec, K. Zupan, S. Pollak, Pan 2017: Author profiling-gender and language variety prediction., in: CLEF (Working Notes), 2017.

[14] B. Koloski, S. Pollak, B. Skrlj, Know your neighbors: Efficient author profiling via follower tweets., in: CLEF (Working Notes), 2020.

[15] B. Koloski, S. Pollak, B. Skrlj, Multilingual detection of fake news spreaders via sparse matrix factorization., in: CLEF (Working Notes), 2020.

[16] A. Basile, M. Chinea-Rios, A.-S. Uban, T. Müller, L. Rössler, S. Yenikent, B. Chulví, P. Rosso, M. Franco-Salvador, Upv-symanto at erisk 2021: Mental health author profiling for early risk prediction on the internet, Working Notes of CLEF (2021) 21–24.

[17] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. URL: https://aclanthology.org/D14-1162. doi:10.3115/v1/D14-1162.

[18] E. Campillo-Ageitos, H. Fabregat, L. Araujo, J. Martinez-Romo, Nlp-uned at erisk 2021: self-harm early risk detection with tf-idf and linguistic features, Working Notes of CLEF (2021) 21–24.

[19] R. Martínez-Castaño, A. Htait, L. Azzopardi, Y. Moshfeghi, Bert-based transformers for early detection of mental health illnesses, in: K. S. Candan, B. Ionescu, L. Goeuriot, B. Larsen, H. Müller, A. Joly, M. Maistro, F. Piroi, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction, Springer International Publishing, Cham, 2021, pp. 189–200.

[20] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, CoRR abs/1911.02116 (2019). URL: http://arxiv.org/abs/1911.02116. arXiv:1911.02116.

[21] C. Spartalis, G. Drosatos, A. Arampatzis, Transfer learning for automated responses to the bdi questionnaire, Working Notes of CLEF (2021) 21–24.

[22] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: http://arxiv.org/abs/1908.10084.

[23] B. Koloski, T. Stepišnik-Perdih, S. Pollak, B. Škrlj, Identification of covid-19 related fake news via neural stacking, in: T. Chakraborty, K. Shu, H. R. Bernard, H. Liu, M. S. Akhtar (Eds.), Combating Online Hostile Posts in Regional Languages during Emergency Situation, Springer International Publishing, Cham, 2021, pp. 177–188.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[25] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with

the natural language toolkit, " O'Reilly Media, Inc.", 2009.

[26] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[27] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter, CoRR abs/1910.01108 (2019). URL: http://arxiv.org/abs/1910.01108. arXiv:1910.01108.

[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[29] D. E. Losada, F. A. Crestani, A test collection for research on depression and language use, in: CLEF, 2016.