

LauSAn at eRisk 2022: Simply and Effectively Optimizing Text Classification for Early Detection

Andreas Säuberli^{1,2}, Sooyeon Cho^{1,2} and Laura Stahlhut^{1,2}

¹Department of Computational Linguistics, University of Zurich, Switzerland

²All authors contributed equally

Abstract

The goal of early detection tasks at eRisk is to classify social media users as early as possible, based on streams of posts written by those users. We present two simple strategies of adapting standard text classification models in order to optimize them for early detection: concatenating the posts in different ways during training and inference, and continuously moving the decision boundary at inference time. We applied these approaches to two different text classification architectures based on pre-trained language models in eRisk 2022's Task 2 (early detection of depression), and were able to reach top 5 placements in all time-sensitive evaluation metrics. A systematic post-submission ablation study confirmed that both strategies were effective at optimizing for early detection.

Keywords

depression detection, early risk detection, threshold scheduling, natural language processing, social media

1. Introduction

In this paper, we describe our team's participation at the 2022 *Conference and Labs of the Evaluation Forum* (CLEF) eRisk task for early detection of depression (Task 2) [1]. Depressive disorders are common in the general population and associated with burdens such as conflict in private life and an increased risk of suicide. Many cases remain undiagnosed, e.g., due to patient somatization and denial or social stigma [2]. It has been shown that certain patterns in a person's writing can be indicative of depression [3, 4]. Being able to detect depression at an early stage from social media, postings could play a part in enabling more people to get treatment earlier and give social media sites a tool to detect potentially suicidal users. The task of early prediction on social media postings can also be extended to other topics, as can be seen from the other eRisk tasks (e.g. early detection of signs of gambling, signs of self-harm or signs of anorexia).


The aim of this task is to detect signs of depression in posts from social media as early as possible. In the training stage, labeled data from the depression subreddit is available to develop depression detection models. During the test phase, models receive the users' posts one by one in chronological order and have to make a binary decision after each post whether the user is depressed or not. The decision for a particular user cannot be undone later.

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

✉ andreas.saeuberli@uzh.ch (A. Säuberli); sooyeon.cho@uzh.ch (S. Cho); lauracelina.stahlhut@uzh.ch (L. Stahlhut)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The main factor that differentiates this task from typical classification tasks is time-sensitivity, i.e., the necessity to classify a user as depressed as early as possible and not only after seeing the entire post history. In addition to standard classification measures such as precision, recall, and F_1 , eRisk uses several time-sensitive evaluation metrics such as *early risk detection error (ERDE)*, latency, speed and latency-weighted F_1 . See Parapar et al. [5] for a complete description of these metrics.

In this paper, we present methods to adapt standard text classification models in order to optimize for early detection, and show that these can be effectively applied to Task 2 of eRisk 2022. Our main contributions are twofold:

- We experiment with different strategies for concatenating a sequence of posts in order to optimize training for the early detection setting.
- We present threshold scheduling, a method to change the decision boundary over the course of a post history in order to optimize for one of the time-sensitive evaluation metrics.

The remainder of this paper is organized in the following way: In Section 2, we mention some related work on earlier installments of eRisk shared tasks, which inspired our approaches. Section 3 describes the task dataset. Section 4 explains our approaches and models. The experimental setup for the submission of our models is introduced in Section 5. In Section 6, we report and discuss our results on the submitted models and post-submission ablation study. Finally, Section 7 provides a general conclusion and brief outlook on future research.

2. Related work

Task 2 (Early Detection of Depression) of CLEF 2022 eRisk is a continuation of eRisk’s 2017 Task 1 and eRisk 2018’s Task 1 (Losada et al. [6, 7]). Thus, there have been multiple groups that have worked with a subset of the dataset we worked on with the same aim we have. While approaches in the preceding versions of this task were mostly concentrated on feature engineering and the application of various classification models, approaches to related early classification tasks in more recent years were often based on Transformer models and transfer learning.

Examples include un Nisa and Muhammad [8], who applied pre-trained BERT embeddings in combination with logistic regression for early detection of self-harm. For the same task, Martínez-Castaño et al. [9] finetuned various transformer models, and trigger a positive decision when the moving average of the predicted probability reaches a specified threshold within a certain time window in the user’s post history. For early detection of signs of pathological gambling, Bucur et al. [10] finetuned BERT models on single posts, and used aggressive decision boundaries in order to prevent false positives.

These submissions, which are based on binary text classification, had to make use of very high decision boundaries, or limit the time window where a positive decision can be made, in order to avoid low precision when repeatedly classifying the same users at every time step. In our submission, we experiment with slightly different approaches to overcome these challenges.

	Users	Posts
Negative	1,493	986,360
Positive	214	90,222
Total	1,707	1,076,582
Positive ratio	12.5%	8.4%

Table 1

Number of users and posts in positive and negative groups in the dataset. Each users’s writing contains a series of posts in chronological order.

3. Dataset

The data used in eRisk 2017 and 2018 are provided as training data by the organizers. This dataset, which was initially presented in Losada and Crestani [11], was collected from Reddit, and contains a chronological collection of posts (title and content) and comments for each of 1,707 users from wide range of subreddits. Each user is labeled either positive (depressed) or negative (control group), based on whether the user has clearly expressed a depression diagnosis in one of their posts (e.g. ‘*I was diagnosed with depression*’).

Table 1 shows the distribution of labels among posts and users. Each user has between 10 and 2,000 posts (median: 366), and each post (title and text combined) contains between 0 and 8,177 whitespace-delimited tokens (median: 13).

4. Methods

We frame the early detection task as a standard text classification problem, where the decision whether a user is depressed or not is done based on a classification of the user’s post history each time a new post is added. In this section, we describe the two approaches we developed to achieve this, as well as the models we chose for the shared task submission. The code used in our experiments is available on GitHub.¹

4.1. Concatenation strategies

We consider three different strategies to prepare the input data. The simplest way would be to use only the most recent post at each given point in time as input to the model. We call this strategy NO-CONCAT. For instance, if a user has 6 posts $[a, b, c, d, e, f]$ where a is the oldest and f is the most recent post, we will train the model by giving it single posts as input ($[a, b, c, d, e, f]$).

However, our hypothesis is that it is difficult to classify whether someone has depression based on a single post, thus we want to include several posts by concatenating a post with a number of directly preceding posts. We propose two such concatenation strategies, which we name CONCAT1 and CONCAT2.

In CONCAT1, the most recent n posts in the user’s post history are concatenated. If a user has 6 posts, we train the model by giving post concatenations as input ($[a, ab, abc, abcd, bcde, cdef]$),

¹<https://github.com/saeub/eRisk2022-LauSAn>

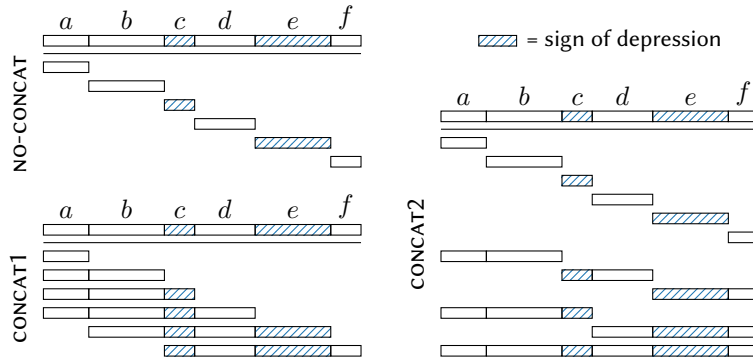


Figure 1: Example of training sample generation for one user with 6 posts using different strategies. Note the different distributions of post length and samples with signs of depression. The oldest post is always to the left. In this example, we assume that this user has 6 posts, $[a, b, c, d, e, f]$. In the NO-CONCAT strategy, each post is fed as a single input. Strategy CONCAT1 concatenates the 4 most recent posts. In the CONCAT2 strategy, 1, 2, 3 and 6 concatenated posts are used as input.

with $n = 4$). A potential problem with this strategy (particularly for users with long post histories) is that almost none of the training samples (only the first $n - 1$) are shorter than n posts, which means that the model will likely perform worse on the first few posts of a user's history. Since it is exactly in this early part where we want to detect most of the depressed users, we propose our final strategy (CONCAT2). This strategy generates both a higher ratio of shorter and additional longer training samples, in order to widen the distribution of the numbers of concatenated posts. For a user's post, we train the model on the individual posts as well as the specified number of concatenations. If we concatenate 1, 2, 3, and 6 posts of a user with 6 posts, the resulting training data is $[a, b, c, d, e, f, ab, cd, ef, abc, def]$ (see Figure 1).

During inference, in CONCAT1 we use the most recent n posts, and in CONCAT2 the entire post history (limited only by the model's maximum input sequence length).

4.2. Threshold Scheduling

As we repeatedly classify users based on their increasingly long post history, taking into account that a positive decision cannot be reversed later, another issue is that a constant decision boundary means that the probability of classifying a user as depressed increases continuously as more posts are processed. Consider the following example: A trained model has a chance $p = 5\%$ of correctly classifying the user as depressed at any point in time. In this case, the chance of detecting a positive user with a history of 10 posts in total is $1 - \prod_{i=1}^{10} (1 - p) = 40\%$, whereas for a history of 100 posts, it is already $> 99\%$. Similarly, the chance of incorrectly classifying a user as depressed increases very quickly. Therefore, users with a long post history have a much higher chance of being misclassified as depressed, while users with a short post history have a much higher chance of remaining undetected. To fix this imbalance, we propose continuously changing the decision boundary during prediction, in order to increase chances of early detection, and reduce misclassifications later in the post history. We use the following exponential threshold scheduling function to achieve this:

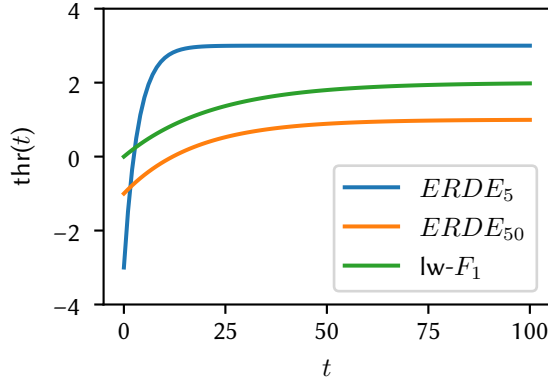


Figure 2: Threshold scheduling functions optimized on the same model for $ERDE_5$, $ERDE_{50}$, and latency-weighted F_1 . In the function optimized for $ERDE_5$, $\text{thr}_{\min} = -3$, $\text{thr}_{\max} = 3$, and $c = 8$.

$$\text{thr}(t) = \text{thr}_{\max} + (\text{thr}_{\min} - \text{thr}_{\max}) \times 10^{-t/c}$$

$\text{thr}(t)$ is the decision boundary applied to the model output after t posts, thr_{\min} is the starting threshold, thr_{\max} is the upper limit threshold, which is asymptotically approached by $\text{thr}(t)$, and c determines how many posts it takes for $\text{thr}(t)$ to reach 90% of the way towards thr_{\max} . We optimize thr_{\min} , thr_{\max} , and c using grid search on the training data, after training the model itself. Figure 2 shows three threshold scheduling functions optimized on the same model for three different metrics. Note that for metrics which highly favor early detection, the initial threshold is much lower, and the threshold increases very quickly.

In addition, we realized that the first few posts are still more difficult to classify, even when using the `CONCAT1` or `CONCAT2` strategies, so we experiment with an additional modification of the function described above, where the first n posts are forced to yield negative decisions, and only then the actual threshold scheduling is applied, i.e., the first n posts are ignored. This is similar to how Martínez-Castaño et al. [9] enforce a minimum number of posts to be read before a positive decision can be made, but we explicitly include n in the grid search space.

4.3. Models

Due to the way we frame the task, any machine learning architecture capable of binary text classification could be used. After initial experiments with different architectures, two models based on pre-trained language models appeared most promising.

The first model is a logistic regression model which we feed a vector representation of the input sequence, obtained by averaging embeddings from the final four layers of a pre-trained BERT model (Devlin et al. 12; `bert-base-uncased` on *Hugging Face*²) across the entire input sequence. This approach is similar to un Nisa and Muhammad [8], although we do not use any additional preprocessing and simply truncate the concatenated raw input posts (title and text) if it is longer than 512 subwords.

²<https://huggingface.co/bert-base-uncased>

For our second model architecture, we finetune a DistilBERT model (Sanh et al. 13; `distilbert-base-uncased` on *Hugging Face*) directly on the binary classification task, again with no additional preprocessing and truncating to 512 subwords.

5. Experimental Setup

For the training of our submitted models, we combined the training and test sets from *eRisk* 2017 and 2018 and split off 20% of the users as validation data for model selection. We undersampled the majority class to reach a positive ratio of 25% in the training data, and trained for three epochs. We used `scikit-learn` [14] for the logistic regression model, and the *Transformers* library by *Hugging Face* [15] for the transformer models.

After experimenting with the different concatenation strategies and threshold scheduling optimizations, we submitted five runs with the following models:

- **LauSAn#0**: Logistic regression with BERT embeddings, NO-CONCAT, exponential threshold scheduling optimized for $ERDE_5$
- **LauSAn#1**: Logistic regression with BERT embeddings, CONCAT1 (5 posts), exponential threshold scheduling optimized for $ERDE_{50}$
- **LauSAn#2**: Logistic regression with BERT embeddings, CONCAT1 (5 posts), exponential threshold scheduling optimized for $ERDE_{50}$ (ignoring the first 3 posts)
- **LauSAn#3**: Logistic regression with BERT embeddings, CONCAT1 (5 posts), exponential threshold scheduling optimized for latency-weighted F_1 (ignoring first 3 posts)
- **LauSAn#4**: Finetuned DistilBERT, CONCAT2, exponential threshold scheduling optimized for $ERDE_5$

For models with CONCAT1, we chose the maximum number of concatenated posts to be $n = 5$. For CONCAT2, we concatenated 1, 2, 3, 4, 10, 20, 30, 40, and 50 posts without overlap. In all cases, we concatenated the posts in reverse order, such that sequences longer than 512 subwords are truncated on the oldest posts rather than the most recent ones.

6. Results and discussion

6.1. Submitted models

Table 2 shows the results of our submitted models on the official test set.³

Out of 62 runs by 13 teams, our runs 4 and 0 ranked 1st and 2nd in terms of $ERDE_5$, run 2 ranked 4th in terms of $ERDE_{50}$, and run 3 ranked 5th in terms of latency-weighted F_1 . However, none of our architectures ranked in the top 10 for several of these metrics at once. This suggests that our threshold scheduling approach was very effective at optimizing towards a specific evaluation metric without re-training the model (note that the models from runs 1, 2, and 3 use exactly the same parameters, apart from the decision boundary).

³The organizers also report ranking-based evaluation metrics (ranking based on model scores after 1, 100, 500, and 1000 posts). Since our model classifies only based on the local post history and does not accumulate scores across time, these measurements are not informative in our case, and we do not report them here.

Run	Precision \uparrow	Recall \uparrow	F_1 \uparrow	$ERDE_5$ \downarrow	$ERDE_{50}$ \downarrow	latency \downarrow	speed \uparrow	lw- F_1 \uparrow
LauSAn#0	0.137	0.827	0.235	0.041	0.038	1	1.000	0.235
LauSAn#1	0.165	0.888	0.279	0.053	0.040	2	0.996	0.278
LauSAn#2	0.174	0.867	0.290	0.056	0.031	4	0.988	0.287
LauSAn#3	0.420	0.643	0.508	0.059	0.041	6	0.981	0.498
LauSAn#4	0.201	0.724	0.315	0.039	0.033	1	1.000	0.315
UNSL#2	0.4	0.755	0.523	0.045	0.026	3	0.992	0.519

Table 2

Test set results of our submitted models, compared with a strong model by team UNSL, which performed well across all time-sensitive metrics.

$ERDE$ is a measure that penalizes late decision. $ERDE_5$ begins punishing strongly from the 5th post, $ERDE_{50}$ from the 50th post. This means that our models were successful at classifying users early, especially runs 0 and 4. Comparing with standard classification metrics, it can be seen that there is a tradeoff between accurate and early classification. For instance, among our models, run 3 achieved the best F_1 as well as the worst $ERDE_5$ and $ERDE_{50}$.

6.2. Post-submission ablation study

The results described above are not very informative for comparing the different parts of our approach independently. Therefore, we conduct a post-submission ablation study in order to assess the contributions of concatenation strategies, threshold scheduling, and model architectures separately, and investigate their interactions. We trained models to cover the full combination space of 2 model architectures, 3 concatenation strategies, and 2 thresholding strategies, each optimized for $ERDE_5$, $ERDE_{50}$, and latency-weighted F_1 . We finetuned the DistilBERT models for 6 epochs in the case of NO-CONCAT and CONCAT1, and 3 epochs in the case of CONCAT2, in order to account for the different number of training samples generated by the processing strategies, without undersampling the training set. Otherwise, the experimental setup is the same as in Section 5. Results can be seen in Table 3. Note that these scores are not directly comparable to the ones in Table 2, as they are only based on our own development set and not the official test set.

In almost all cases, threshold scheduling with the exponential function defined in Section 4.2 outperforms or equals the constant decision boundary. These results confirm the hypothesis that a constant threshold is suboptimal for repeatedly classifying the same users. The optimized threshold scheduling functions look similar across different concatenation strategies: starting below the midpoint, and quickly increasing towards a value above it.⁴ In contrast, the optimal constant threshold fluctuates strongly between optimization metrics, and in one case even leads to an F_1 score of 0.

Regarding the concatenation strategies, the picture is less clear. $ERDE_{50}$ and F_1 both profit from the concatenation strategies that involve history (CONCAT1, CONCAT2). Compared to NO-CONCAT, they tend to lose performance when optimized for $ERDE_5$, although CONCAT2

⁴An exception is the model trained with CONCAT2 and threshold scheduling optimized for latency-weighted F_1 , where the optimized exponential function turns out to be equal to the constant one.

Logistic regression with averaged BERT embeddings:

Threshold scheduling	Concatenation strategy								
	NO-CONCAT			CONCAT1			CONCAT2		
	$ERDE_5$	$ERDE_{50}$	$lw-F_1$	$ERDE_5$	$ERDE_{50}$	$lw-F_1$	$ERDE_5$	$ERDE_{50}$	$lw-F_1$
constant, optimized for									
... $ERDE_5$	0.126	0.126	0.000	0.117	0.101	0.458	0.106	0.075	0.512
... $ERDE_{50}$	0.132	0.089	0.388	0.124	0.075	0.419	0.106	0.075	0.512
... latency-weighted F_1	0.124	0.093	0.485	0.113	0.090	0.490	0.104	0.077	0.528
exponential, optimized for									
... $ERDE_5$	0.073	0.069	0.347	0.081	0.079	0.326	0.073	0.066	0.361
... $ERDE_{50}$	0.086	0.068	0.470	0.077	0.054	0.413	0.087	0.068	0.471
... latency-weighted F_1	0.086	0.068	0.470	0.105	0.065	0.521	0.104	0.077	0.528

Finetuned DistilBERT:

Threshold scheduling	Concatenation strategy								
	NO-CONCAT			CONCAT1			CONCAT2		
	$ERDE_5$	$ERDE_{50}$	$lw-F_1$	$ERDE_5$	$ERDE_{50}$	$lw-F_1$	$ERDE_5$	$ERDE_{50}$	$lw-F_1$
constant, optimized for									
... $ERDE_5$	0.129	0.109	0.224	0.174	0.094	0.256	0.150	0.097	0.267
... $ERDE_{50}$	0.167	0.106	0.244	0.171	0.099	0.251	0.151	0.083	0.308
... latency-weighted F_1	0.174	0.114	0.261	0.173	0.096	0.258	0.158	0.085	0.315
exponential, optimized for									
... $ERDE_5$	0.086	0.083	0.391	0.105	0.102	0.272	0.094	0.088	0.353
... $ERDE_{50}$	0.100	0.096	0.318	0.103	0.086	0.362	0.096	0.052	0.474
... latency-weighted F_1	0.088	0.083	0.404	0.107	0.102	0.271	0.102	0.057	0.478

Table 3

Ablation results for logistic regression with averaged BERT embeddings. The changing factors were concatenation strategy (NO-CONCAT, CONCAT1, CONCAT2), threshold scheduling (constant, exponential), and the metric for which we optimized ($ERDE_5$, $ERDE_{50}$, latency-weighted F_1), which resulted in 18 trained models for each architecture. Best scores for each metric and architecture are shown in bold.

manages to recover some of it, likely due to the additional short samples seen during training. Overall, it appears that CONCAT2 leads to better results, unless the evaluation metric exclusively favors very early detection as in $ERDE_5$.

Comparing the two model architectures, the logistic regression model mostly outperforms DistilBERT, especially with NO-CONCAT and CONCAT1 strategies. DistilBERT outperforms the logistic regression model in terms of $ERDE_5$ and $ERDE_{50}$ only in the setting with exponential threshold scheduling and CONCAT2. This suggests that classifying averaged word representations from pre-trained language models can be a viable option for this task. An explanation for this could be that depression mainly manifests itself in the general semantic topic of the posts rather than subtle linguistic details, and can thus largely be captured by averaged word embeddings.

7. Conclusion and outlook

We presented our approaches to 2022’s eRisk task for early detection of depression. Framing this problem as a modified text classification, we experimented with different strategies for input concatenation and threshold scheduling in order to take the post history into account and enable early detection. Two of our models ranked first and second in the $ERDE_5$ metric, which implies that they were particularly successful at detecting depression as early as possible.

In addition, we conducted an ablation study in order to determine the effect of the concatenation strategies and threshold scheduling. Our results show that adapting the classification threshold at inference time (specifically, starting with a small threshold and increasing over time) is highly effective for supporting early detection when the total number of samples is unknown at first. We tested three concatenation strategies and observed that our CONCAT2 strategy, which generates training data with both short and long post histories, results in a good trade-off between early and accurate detection. We also show that for this specific task, classification of pre-trained hidden representations with traditional machine learning models can be a very effective and more robust alternative to finetuning.

Overall, the performance achieved on early detection of depression still leaves room for improvement, with best achieved F_1 scores in eRisk 2022 of 0.712 by NLPGroup-IISERB. In future experiments, our approaches could be extended by further optimizing hyperparameters for the concatenating strategies. Comparing our method of using local classification scores with threshold scheduling with more globally accumulated confidence scores may provide more insights into its effectiveness. And finally, our approaches leave room for integrating domain-specific knowledge such as engineered linguistic features, which we have not explored at all.

Acknowledgement

We would like to thank Simon Clematide and Andrianos Michail for their valuable feedback and to the Department of Computational Linguistics for providing the technical infrastructure for our experiments.

References

- [1] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, eRisk 2022: Pathological gambling, depression, and eating disorder challenges, in: European Conference on Information Retrieval, Springer, 2022, pp. 436–442.
- [2] L. Goldman, N. Nielsen, H. Champion, Awareness, diagnosis, and treatment of depression, *Journal of General Internal Medicine* 14 (2001) 569 – 580. doi:10.1046/j.1525-1497.1999.03478.x.
- [3] J. W. Pennebaker, M. R. Mehl, K. G. Niederhoffer, Psychological aspects of natural language use: Our words, our selves, *Annual Review of Psychology* 54 (2003) 547–577. URL: <https://doi.org/10.1146/annurev.psych.54.101601.145041>. doi:10.1146/annurev.psych.54.101601.145041.

arXiv:<https://doi.org/10.1146/annurev.psych.54.101601.145041>, PMID: 12185209.

- [4] S. R. Brown, W. Weintraub, Verbal behavior: Adaptation and psychopathology, *Political Psychology* 5 (1984) 107. doi:10.2307/3790837.
- [5] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Overview of eRisk 2021: Early risk prediction on the internet, in: *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, 2021, pp. 324–344.
- [6] D. Losada, F. Crestani, J. Parapar, eRisk 2017: CLEF lab on early risk prediction on the internet: Experimental foundations, 2017, pp. 346–360. doi:10.1007/978-3-319-65813-1_30.
- [7] D. E. Losada, F. Crestani, J. Parapar, Overview of eRisk: early risk prediction on the internet, in: *International Conference of the Cross-language Evaluation Forum for European Languages*, Springer, 2018, pp. 343–361.
- [8] Q. un Nisa, R. Muhammad, Towards transfer learning using BERT for early detection of self-harm of social media users, in: *Proceedings of the Working Notes of CLEF 2021 – Conference and Labs of the Evaluation Forum*, September 21-24, 2021, 2021.
- [9] R. Martínez-Castaño, A. Htait, L. Azzopardi, Y. Moshfeghi, Early risk detection of self-harm and depression severity using BERT-based transformers, *Working Notes of CLEF (2020)* 16.
- [10] A.-M. Bucur, A. Cosma, L. P. Dinu, Early risk detection of pathological gambling, self-harm and depression using BERT, arXiv preprint arXiv:2106.16175 (2021).
- [11] D. Losada, F. Crestani, A test collection for research on depression and language use, volume 9822, 2016, pp. 28–39. doi:10.1007/978-3-319-44564-9_3.
- [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] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).
- [14] 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.
- [15] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.