Early Risk Detection of Self-Harm and Depression Severity using BERT-based Transformers

iLab at CLEF eRisk 2020

Rodrigo Martínez-Castaño[1,2], Amal Htait[2],
Leif Azzopardi[2], and Yashar Moshfeghi[2]

[1] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS),
Universidade de Santiago de Compostela, Spain
`rodrigo.martinez@usc.es`
[2] Department of Computer and Information Sciences, University of Strathclyde, UK
{`amal.htait,leif.azzopardi,yashar.moshfeghi`}`@strath.ac.uk`

Abstract. This paper briefly describes our research groups' efforts in tackling Task 1 (Early Detection of Signs of Self-Harm), and Task 2 (Measuring the Severity of the Signs of Depression) from the CLEF eRisk Track. Core to how we approached these problems was the use of BERT-based classifiers which were trained specifically for each task. Our results on both tasks indicate that this approach delivers high performance across a series of measures, particularly for Task 1, where our submissions obtained the best performance for precision, F1, latency-weighted F1 and ERDE at 5 and 50. This work suggests that BERT-based classifiers, when trained appropriately, can accurately infer which social media users are at risk of self-harming, with precision up to 91.3% for Task 1. Given these promising results, it will be interesting to further refine the training regime, classifier and early detection scoring mechanism, as well as apply the same approach to other related tasks (e.g., anorexia, depression, suicide).

Keywords: Self-Harm · Depression · Classification · Social Media · Early Detection · BERT · XLM-RoBERTa

1 Introduction

The eRisk CLEF track aims to explore the development of methods for early risk detection on the Internet, their evaluation, and the application of such methods

for improving the health and well being of individuals [8–11]. Early detection technologies can be employed in different areas, particularly those related to health and safety. For instance, in [9] they examined whether it was possible to identify grooming activities of paedophiles given posts to online forums. While in [10, 11], they explored whether it was possible to detect users that were depressed or anorexic from their posts, and crucially how quickly this could be detected. This year the focus is on detecting the early signs of self-harm from people's posts to social media (Task 1), and whether it is possible to infer how depressed people are given such posts (Task 2) [12]. Below is an elaborated description of each task.

Task 1: Early Detection of Signs of Self-Harm. This first task consists of triggering alerts for users that present early signs of committing self-harm. A tagged set of users and their posts to Reddit[3] groups was provided for training purposes. The different methods were benchmarked using a system that simulates a real-time scenario introduced in [11]. The posts from the users of the test dataset are served in rounds, one post at a time (simulating their live posting to the Reddit groups). The task then is to provide a decision about each user given their posts, and to do so as early as possible (i.e., with the fewest posts). For the evaluation, the correctness of the prediction (i.e., whether the user will cause self-harm or not) is not the only factor taken into account, but also the delay taken to emit the alerts. Clearly, the sooner a person who is likely to self-harm is identified, the sooner the intervention can be provided.

Task 2: Measuring the Severity of the Signs of Depression. This task consists of automatically estimating the level of several symptoms associated with depression. For that, a questionnaire with 21 questions related to different feelings and well-being (e.g., sadness, pessimism, fatigue) is provided. Each question has between four and seven possible answers which are related to different levels of severity (or relevance) of the symptom or behaviour. A sample of users with their answers to the questionnaire and their writings at Reddit was given. To benchmark the different approaches, a new set of users and their writings is provided, for which every team has to predict their answers.

Thus, the goal of this paper is to explore the potential of a BERT-based classifier coupled with a novel scoring mechanism for the early detection of self-harm and depression. This paper is structured as follows. In Section 2 we describe our general approach for both tasks by using BERT-based models for sentence classification. In Section 3 and Section 4 we explain how the classifiers were trained and applied for Task 1 and Task 2 respectively. Section 5 covers the analysis of our results, where our approach performs the best across a number of metrics for both tasks. Finally, in Section 6 we summarise the contributions of these working notes.

---

[3] https://reddit.com/

2 Approach

A breakthrough in the use of machine learning for Natural Language Processing (NLP) appeared with the generative pre-training of language models on a diverse corpus of unlabelled text, such as ELMo [15], BERT [4], OpenAI GPT [16], XLM [6], and RoBERTa [7]. Such a technique demonstrated large gains on a variety of NLP tasks (e.g., sequence or token classification, question answering, semantic similarity assessment, document classification). In particular, BERT (Bidirectional Encoder Representations from Transformers) [4, 3], the model by Google AI, proved to be one of the most powerful tools for text classification [13, 14, 5]. BERT is based on the Transformer architecture [18] and it was trained for both masked word prediction and next sentence prediction at the same time. As input, BERT takes two concatenated segments of text which are delimited with special tokens and whose length respects a defined maximum. The model was pre-trained on a huge dataset of unlabelled text. It is typically used within a text classifier for sentence tokenisation and text representation. A standard BERT classifier is presented in Figure 1 where a sentence is tokenised, represented in embeddings and then classified. The results are normalised between 0 and 1 using the softmax function, representing the probability of the *input* sentence to belong to a certain class (e.g., the probability of the sentence to be written by a self-harmer).
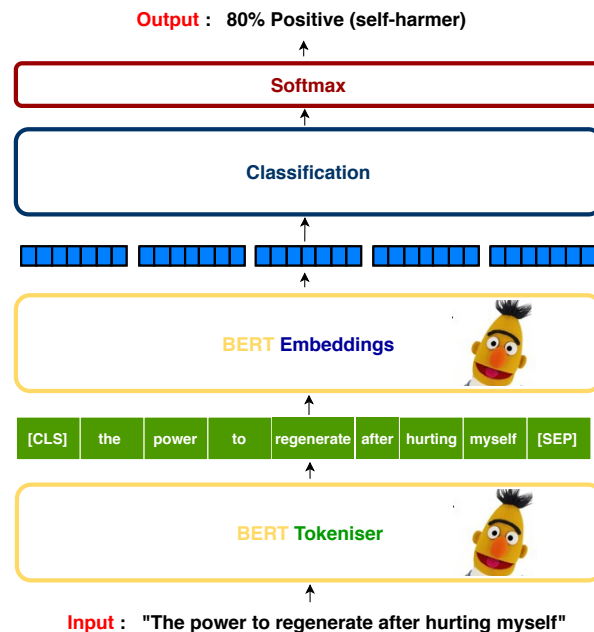

Fig. 1. BERT-based Classification Architecture.

As for RoBERTa [7] (a replication study of BERT pre-training by Facebook AI), it shares a similar architecture with BERT but with a different pre-training approach. RoBERTa was trained over ten times more data, the next sentence prediction objective was removed, and the masked word prediction task was improved with the introduction of a dynamic masking pattern applied to the training data.

In another attempt to improve the language model, Facebook AI presented XLM-RoBERTa [2] with the pre-training of *multilingual* language models. This new improvement led to significant performance gains in text classification. For our participation at the eRisk challenges of 2020, variety of pre-training language models were tested: BERT, DistillBERT, RoBERTa, and XLM-RoBERTa, among others. However, the best performance was achieved when using XLM-RoBERTa on our training data. In our work, we used Ernie [4], a Python library for sentence classification built on top of Hugging Face Transformers [5], the main library that implements state-of-the-art general-purpose Transformer-based architectures.

Most of the pre-training language models, including XLM-RoBERTa, have a maximum input length of 512 tokens. In our work, we experimented with input sentences of sizes between 32 and 128 tokens due to GPU memory restrictions. The best results were achieved with an input size of 128 tokens. Note that Reddit posts are usually shorter than 128 tokens. Therefore, using an input size larger than 128 would not substantially increase performance, but it would significantly increase the required computational resources. In the few cases where the Reddit posts were longer, we split them based on punctuation marks in an attempt to respect the context of the writings posted by the users. When training the classifiers, the weights of the pre-trained base models (e.g., XLM-RoBERTa) are updated, in addition to the classification head.

For our participation at the eRisk challenges of 2020, both Task 1 and Task 2, we used the previously explained approach for sentence classification. However, in each task, the employed training schedule and training data were varied and tailored to fit the task scenarios, as explained in the following sections.

3 Task 1 - Early Risk Detection of Self-Harm

We trained a number of different language models based on the original BERT architecture with a classification head to predict whether a sentence was written by a subject that self-harms or not. Those models are the base to predict if a user is likely to self-harm and thus, triggering an alert, given a stream of texts. All of our final models were based on XLM-RoBERTa, which demonstrated better performance for this task.

---

[4] https://github.com/labteral/ernie/
[5] https://github.com/huggingface/transformers/

3.1 Data

To train our models, we avoided using the training dataset provided by the eRisk organisers for two reasons. First, during the beginning of our experimentation, we found that the results obtained with our BERT-based approach were not promising enough to beat the existing approaches used in 2019. Second, the training dataset matches the test data of the eRisk 2019's task. Taking it out from the training stage led us to be able to compare our results with the obtained by the last year's participants in our search for models with greater performance.

The data collected and used for training our models were obtained from the Pushshift Reddit Dataset [1] through its public API[6], which exposes a repository with constantly updated and almost complete dataset of all the public Reddit data. We downloaded all the available submissions and comments written to the most popular subreddit about self-harm (r/selfharm). From those posts, we extracted $42,839$ authors. In addition, we collected all of the posts in any other subreddit for those authors (SELFHARM-USERS-TEXTS dataset). Then, we obtained an equivalent amount of random users from which we also extracted all their posts (RANDOM-USERS-TEXTS dataset). We filtered the obtained datasets in several ways. First, we checked that there were not any user collision between the two collections. After identifying some of the main self-harm related subreddits (r/selfharm, r/Cutters, r/MadeOfStyrofoam, r/SelfHarmScars, r/StopSelfHarm, r/CPTSD and r/SuicideWatch), we removed the users from RANDOM-USERS-TEXTS having at least one post in any of them. All the users with more than $5,000$ submissions were removed since those with an extremely high number of posts seem more likely to be bots. Besides, the vast majority of the users had posted fewer times so we presumed to have more chances to profile the average user below that threshold. We also pruned the less active users under 50 submissions. The number of sentences was expanded by splitting the users' texts that were too long for the parameters we utilised in our models. Otherwise, the sentences would be truncated during training, potentially losing valuable information. We split the large posts into groups of contiguous sentences of approximately the maximum length in tokens utilised in our models and following the punctuation marks hierarchy (e.g., prioritising the splits on full stops over commas). As commented before, a maximum length of 128 tokens was set so the models could be fine-tuned in commercial GPUs.

We created several datasets mainly derived from SELFHARM-USERS-TEXTS and RANDOM-USERS-TEXTS for training our model candidates. These datasets are presented in Table 1, and explained below:

– A manually created dataset:
  • REAL-SELFHARMERS-TEXTS: This dataset was created with the aim of obtaining a bigger but similar dataset to the one provided by the eRisk organisers. We manually tagged 354 users as real self-harmers from the users of the SELFHARM-USERS-TEXTS dataset. Then, we filtered the last

---

[6] https://pushshift.io/api-parameters/

$1,000$ submissions and comments for every user. We also pruned the writing sequences just before their first writing at r/selfharm. After that, we filtered the users with at least 10 writings remaining, ending up with a total of 120 real self-harmers. For the negative class, we took a sample of random users from the dataset RANDOM-USERS-TEXTS in the same proportion as in the provided training data: ~ 7.3 random users per self-harmer.

– Datasets automatically generated from SELFHARM-USERS-TEXTS and RANDOM-USERS-TEXTS after removing the users from REAL-SELFHARMERS-TEXTS. In Figure 2, we show the distribution of posts per user for the original datasets (SELFHARM-USERS-TEXTS and RANDOM-USERS-TEXTS) and the derived ones utilised to train the final classifiers:

- USERS-TEXTS-200K: This dataset was generated by random sampling $200K$ writings from both SELFHARM-USERS-TEXTS (as self-harmers) and RANDOM-USERS-TEXTS (as non self-harmers), with $100K$ from each dataset. Note that we experimented by replicating last years' task with different sizes of sampling such as $2K$, $20K$, $100K$, $300K$, $400K$ and $500K$ writings, but the best results were achieved with a sampling size of $200K$ writings.

- USERS-TEXTS-2M: This dataset is a variant of USERS-TEXTS-200K; a balanced dataset with ten times more sentences, totalling $2M$ writings. Note that, during our experimentation replicating last years' task, using a training set larger than $200K$ did not improve the results except for the $ERDE_5$ metric with the $2M$ writings.

- USERS-SUBMISSIONS-200K: This dataset was generated in a similar procedure as USERS-TEXTS-200K, with $200K$ random sampled writings, but with the difference of avoiding comments. Therefore, sampling users' submissions exclusively.

Dataset	Class	Users	Subreddits	Sentences	Years
REAL-SELFHARMERS-TEXTS	selfharm	120	$1,346$	$8,943$	2013 - 2020
	random	875	$5,585$	$87,260$	2009 - 2020
USERS-TEXTS-200K	selfharm	$9,487$	$9,797$	$107,277$	2006 - 2020
	random	$14,280$	$9,793$	$107,152$	2006 - 2020
USERS-TEXTS-2M	selfharm	$10,454$	$26,931$	$1,075,476$	2006 - 2020
	random	$17,548$	$26,409$	$1,076,707$	2005 - 2020
USERS-SUBMISSIONS-200K	selfharm	$10,319$	$13,681$	$131,233$	2006 - 2020
	random	$15,937$	$14,913$	$128,064$	2005 - 2020

Table 1. Some statistics of the datasets used to train the classifiers.
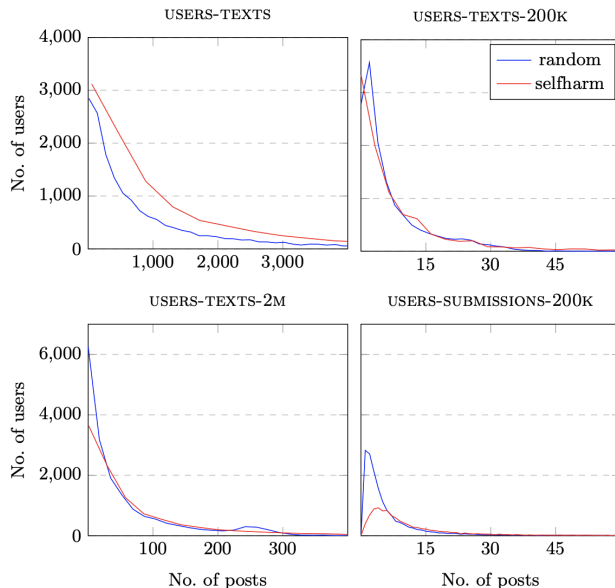
Fig. 2. Distribution of the number of posts per user in the datasets SELFHARM-USERS-TEXTS, RANDOM-USERS-TEXTS and the derived datasets from them.

3.2 Method

For our participation in Task 1 of eRisk we trained three models for binary sentence classification, all of them based on the XLM-RoBERTa-base language model (since it behaved better than other variants we tried such as BERT, DistillBERT, XLNet, etc.):

– XLMRB-SELFHARM-200K trained with the dataset USERS-TEXTS-200K.
– XLMRB-SELFHARM-2M trained with the dataset USERS-TEXTS-2M.
– XLMRB-SELFHARM-SUB-200K trained with the dataset USERS-SUBMISSIONS-200K.

We established for those models a maximum length of tokens as 128 per sentence, a training rate of $2e-5$ and a validation size of the 20%.

In order to predict if a user has or has not risk of self-harm, we averaged the predicted probability of the known writings for every user. We omitted the prediction of sentences with less than 10 tokens as we concluded that the performance on smaller sentences is poor. Since the provided training set was the test set of the last year's task, we used it to compare the performance of our models with the participants of the previous year. We defined several parameters to determine if the system should trigger an alert given a list of known user's texts:

the minimum average probability threshold (θ), the minimum number of texts necessary to trigger an alert, and the maximum number of texts that the system will take into account to make its decisions on the subjects. Given a growing list of texts from a user, the system will trigger an alert if the average probability of the known texts for that user is greater or equal than θ, the number of known texts is greater or equal to the minimum, and lower or equal to the maximum.

The parameters were adjusted in five variants by finding their optimal values for F1 and the eRisk related metrics: latency-weighted F1, ERDE$_5$ and ERDE$_{50}$ with the REAL-SELFHARMERS-TEXTS dataset. For example, in Figure 3 it can be observed that the best value for latency-weighted F1 with any θ is obtained when waiting for at least 10-12 texts for XLMRB-SELFHARM-200K. We chose the model with the best performance for each target metric. The selected parameters for each variant can be observed in Table 2 and the results obtained with the REAL-SELFHARMERS-TEXTS dataset are shown in Table 3.

After choosing the parameters with the REAL-SELFHARMERS-TEXTS dataset, we tested the classifiers with the last year's test data for the same task as showed in Table 4, where we compare the obtained results with the best performer of 2019 for that task: UNSL. That team obtained the best results for precision, F1, ERDE$_5$, ERDE$_{50}$ and latency-weighted F1. With the classifiers that we used in our submission, we improved their results for F1, ERDE$_5$, ERDE$_{50}$ and latency-weighted F1.

Run	Model	Target Metric	θ	Min. posts	Max. posts
0	XLMRB-SELFHARM-200K	*latency-weighted F1*	0.75	10	50
1	XLMRB-SELFHARM-2M	*latency-weighted F1*	0.76	10	50
2	XLMRB-SELFHARM-2M	*ERDE_5*	0.69	2	5
3	XLMRB-SELFHARM-SUB-200K	*ERDE_50*	0.64	45	45
4	XLMRB-SELFHARM-200K	*F1*	0.68	100	100

Table 2. Combinations of models and parameters for the five submitted runs.

4 Task 2

4.1 Data

For our participation in Task 2 of eRisk, we used the training dataset provided by the task's organisers. Both training and test datasets consist of Reddit posts written by users who have answered the questionnaire. The training dataset includes a total of $10,941$ posts by 20 users, and the test dataset includes $35,562$ posts by 70 users.
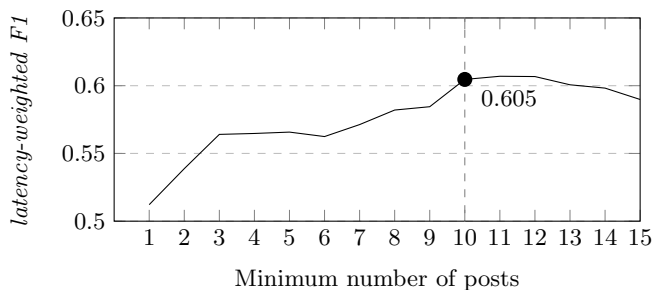
Fig. 3. Latency-weighted F1 when varying the minimum number of texts to trigger an alert. Model XLMRB-SELFHARM-200K with the REAL-SELFHARMERS-TEXTS dataset. A maximum of 50 posts are taken into account.

Run	P	R	F1	ERDE 5	ERDE 50	Latency TP	Speed	Latency-weighted F1
0	0.646	0.608	0.627	0.125	0.052	10	0.965	**0.605**
1	**0.736**	0.533	0.618	0.123	0.059	10	0.965	0.597
3	0.401	0.708	0.517	**0.057**	0.050	2	0.050	0.515
4	0.350	**0.825**	0.491	0.143	**0.044**	45	0.830	0.408
5	0.720	0.600	**0.655**	0.124	0.124	100	0.632	0.414

Table 3. Results obtained by our five final variants with the REAL-SELFHARMERS-TEXTS dataset when using the optimal parameters.

Team	Run	P	R	F1	ERDE 5	ERDE 50	Latency TP	Speed	Latency-weighted F1
UNSL 2019	0	**0.71**	0.41	0.52	0.090	0.073	2	1	0.52
UNSL 2019	4	0.31	**0.88**	0.46	0.082	0.049	3	.99	0.45
iLab	0	0.68	0.66	**0.67**	0.125	0.046	10	0.97	**0.64**
iLab	1	0.69	0.59	0.63	0.124	0.054	10	0.97	0.61
iLab	2	0.33	0.71	0.45	**0.062**	0.057	2	1	0.44
iLab	3	0.34	0.83	0.48	0.144	**0.045**	45	0.83	0.40
iLab	4	0,68	0.66	**0.67**	0.125	0.125	100	0.63	0.42

Table 4. Results obtained by our five final variants with the 2019 dataset compared to the results obtained by UNSL.

An analogous approach as the one employed for Task 1, with random posts from users connected solely by a common subreddit, was not possible this time. Therefore, and due to the small dataset for training (only 20 different users), we used the full provided training dataset in order to train the classifiers. For each question of the questionnaire, we modified the training dataset by assigning the same class to all the texts posted by a given user (i.e., each class matches one of the available answers). Thus, we obtained a different training set for each question of the questionnaire, and, therefore, one different multi-class classifier.

4.2 Method

For this task, we applied a similar method as the one employed in Task 1, but we treated the problem as a multi-class labelling problem. We created three variants, only differing in the base language model and the pre-processing of the training data, as it can be observed in Table 5. For the runs 1 and 2, we expanded the training by splitting texts larger than 128 tokens in the same way as in Task 1. However, for Run 3, sentences larger than 128 tokens were truncated during the training phase.

Run	Base LM	Strategy
1	XLM-RoBERTa-base	split
2	RoBERTa-base	split
3	RoBERTa-base	truncate

Table 5. Base language models and training set variants used for Task 2.

For each variant, we fine-tuned the base language model with a head for multi-class classification for every question. As shown in Table 6, we balanced the class weights of every question model for all the variants. The RoBERTa-based classifiers were trained for 4 epochs, whereas we executed 5 epochs for the XLM-RoBERTa-based ones. Those numbers of epochs were found to be optimal in all the models we created during our experimentation for Task 1. We established the maximum sentence length to 128 tokens and the learning rate to $2e-5$ to train all the models. We assigned a 20% of the training data for validation.

For a given user and variant, we predict the questionnaire answer in the following way: given a question and the associated classifier, we obtain the *softmax* prediction vector for every text written by that user and we sum them. The class with the highest accumulated value is the answer to the questionnaire we predict. As in Task 1, during prediction, if the input texts are larger than 128 tokens, we split them and average the predictions of the chunks.

Question	Answers						
	0	1 (1a)	2 (1b)	3 (2a)	2b	3a	3b
1	1.000	1.079	14.399	0.000	-	-	-
2	1.000	1.291	9.003	1.935	-	-	-
3	1.000	2.151	1.956	4.001	-	-	-
4	2.660	1.000	4.523	247.375	-	-	-
5	1.000	2.332	220.406	2.751	-	-	-
6	1.000	93.442	9.658	62.294	-	-	-
7	1.000	2.630	2.820	2.707	-	-	-
8	1.000	1.084	6.410	5.449	-	-	-
9	1.000	1.223	5.250	0.000	-	-	-
10	1.000	10.119	3.634	30.020	-	-	-
11	1.000	1.981	1.548	3.308	-	-	-
12	1.000	1.332	41.777	2.451	-	-	-
13	1.000	10.514	10.041	5.239	-	-	-
14	1.000	3.386	3.300	6.204	-	-	-
15	1.622	3.072	1.000	5.106	-	-	-
16	1.000	9.361	7.254	2.280	1.439	0.000	1.261
17	1.076	1.000	2.042	0.000	-	-	-
18	1.000	3.427	1.584	18.572	189.781	20.243	18.021
19	1.000	2.531	1.102	20.576	-	-	-
20	1.594	1.000	6.826	4.764	-	-	-
21	1.170	1.000	1.790	3.227	-	-	-

Table 6. Class weights for each question used to train the classifiers in all the variants.

5 Results

Table 7 shows the performance of our runs for Task 1, while Table 8 shows the performance of our runs for Task 2. In each table, the best scores among all the participants are highlighted in bold. Other runs from other teams have also been included to show the best performing runs for each task on each metric.

For task 1, the evaluation metrics used were [11]:

- The standard classification measures **precision (P)**, **recall (R)** and **F1**, are computed with respect to the positive class, since they are the only cases that trigger alerts.
- **ERDE** (Early Risk Detection Error) [8], is an error measure that introduces a penalty for late correct alerts (true positives) and depends on the number of user writings seen before the alert. Two sets of user writing numbers are taken into consideration in this challenge: 5 and 50. Contrary to the other metrics, the lower the value of ERDE, the better the performance of the system.
- **Latency**$_{TP}$ measures the delay in detecting true positives, defined as the median number of writings used to detect positive cases.
- **Speed** is the system's overall speed factor, where it will be equal to 1 for a system whose true positives are detected right at the first writing, and almost 0 for a slow system, which detects true positives after hundreds of writings.
- **Latency-weighted F1** [17] score is equal to $F1 \cdot speed$, and a perfect system gets latency-weighted F1 equals to 1.

For Task 2, the following metrics were used [11]:

- **AHR** (Average Hit Rate) is the average of Hit Rate (HR) across all users, and HR is the ratio of cases where the automatic questionnaire has exactly the same answer as the actual questionnaire.
- **ACR** (Average Closeness Rate) is the average of Closeness Rate (CR) across all users, and CR is equal to *(mad - ad)/mad*, where *mad* is the maximum absolute difference, which is equal to the number of possible answers minus one, and *ad* is the absolute difference between the real and the automated answer.
- **ADODL** (Average DODL) is the averaged of Difference between Overall Depression Levels (DODL) across all users. DODL computes the overall depression level (sum of all the answers) for the real and automated questionnaire and, next, the absolute difference (ad_overall) between the real and the automated score is computed. DODL is normalised into [0,1] as follows: *DODL = (63 - ad_overall)/63*.
- **DCHR** (Depression Category Hit Rate) computes the fraction of cases where the automated questionnaire led to a depression category (out of 4 categories: nonexistence, mild, moderate and severe) that is equivalent to the depression category obtained from the real questionnaire.

Run	P	R	F1	ERDE 5	ERDE 50	Latency TP	Speed	Latency-Weighted F1
0	0.833	0.577	0.682	0.252	0.111	10	0.965	**0.658**
1	**0.913**	0.404	0.560	0.248	0.149	10	0.965	0.540
2	0.544	0.654	0.594	**0.134**	0.118	2	0.996	0.592
3	0.564	0.885	0.689	0.287	**0.071**	45	0.830	0.572
4	0.828	0.692	**0.754**	0.255	0.255	100	0.632	0.476

Table 7. The performance for each run we submitted on Task 1: Early Detection of Signs of Self-Harm. Note that for each bolded metric our run gave the highest performance.

Team	Run	AHR	ACR	ADODL	DCHR
BioInfo@UAVR	0	**38.30%**	69.21%	76.01%	30.00%
prhlt-upv	0	34.01%	67.07%	80.05%	**35.71%**
prhlt-upv	1	34.56%	67.44%	80.63%	**35.71%**
RELAI	0	36.39%	68.32%	**83.15%**	34.29%
iLab	0	36.73%	68.68%	81.07%	27.14%
iLab	1	37.07%	**69.41%**	81.70%	27.14%
iLab	2	35.99%	69.14%	82.93%	34.29%

Table 8. The performance for each run we submitted on Task 2: Measuring the severity of the signs of depression, along with the runs from other teams that scored higher.

For Task 1, our team's performance for each of the key metrics was the best compared to the other teams this year. Given our training schedule which tried to maximise the performance for each metric per run, we can see that no specific run was the best across all the metrics, but rather there is a trade-off between metrics. For example, Run 1 obtains a precision score of 0.913, but has the lowest recall, while Run 4 obtains the highest F1, but not the best precision or recall. Of most interest is the performance on the eRisk-specific metrics, where our runs obtained notably the best results. With Run 0 we obtained a latency-weighted F1 of 0.66, where the second-best result was obtained by the team UNSL with their run 1 at 0.61. For $ERDE_5$, Run 2 scored 0.134, whereas the second-best team was again UNSL with their run 1 at 0.172 (where lower is better). For $ERDE_{50}$, our Run 3 obtained a score of 0.071, whereas all the other runs ranged between 0.11 to 0.25.

For Task 2, our team's performance was the best for ACR, and competitive for the other metrics. For AHR, ADODL and DCHR our performances were within 1-2% of the best performances submitted. Interestingly, while the ADODL scores were around 81-83%, this did not translate into a better classification of depression category as surmised by DCHR, which was 34% at best. This disparity may be due to how we employed the BERT based classifier (i.e., we made separate models to predict the results of each question). However, it may be more appropriate to jointly predict the results of all questions and the final depression category. This is because the questions will have a high correlation between answers, and information for inferring the answer for one question, may be useful in inferring others when taken together.

6 Summary

In this paper we have described how we employed a BERT-based classifier for the tasks of the CLEF eRisk Track: Task 1, early risk detection of self-harm; and Task 2, inferring answers to a depression survey. Our results on both tasks indicated that this approach works very well and obtains very good performance (the best on Task 1 and very competitive performance on Task 2). These results are perhaps not too surprising, given the impact that BERT-based models have been making in improving many other tasks. However, a key difference in this work is how we trained the model. In future work, we will explore and compare different training schedules and classifiers extensions for these tasks, but also for other related tasks (e.g., classifying whether someone is like to suffer from anorexia, depression).

Acknowledgements

References

1. Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., Blackburn, J.: The pushshift reddit dataset. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 14, pp. 830–839 (2020)
2. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2019)
3. Devlin, J., Chang, M.W.: Open sourcing bert: State-of-the-art pre-training for natural language processing. Google AI Blog, November **2** (2018)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Gao, Z., Feng, A., Song, X., Wu, X.: Target-dependent sentiment classification with bert. IEEE Access **7**, 154290–154299 (2019)
6. Lample, G., Conneau, A.: Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291 (2019)
7. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
8. Losada, D.E., Crestani, F.: A test collection for research on depression and language use. In: International Conference of the Cross-Language Evaluation Forum for European Languages. pp. 28–39. Springer (2016)
9. Losada, D.E., Crestani, F., Parapar, J.: CLEF 2017 eRisk overview: Early Risk prediction on the internet: Experimental foundations. CEUR Workshop Proceedings **1866** (2017)
10. Losada, D.E., Crestani, F., Parapar, J.: Overview of eRisk 2018: Early Risk Prediction on the Internet (extended lab overview). CEUR Workshop Proceedings **2125** (2018)
11. Losada, D.E., Crestani, F., Parapar, J.: Overview of eRisk 2019 Early Risk Prediction on the Internet. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11696 LNCS**(September), 340–357 (2019)
12. Losada, D.E., Crestani, F., Parapar, J.: Overview of eRisk 2020: Early Risk Prediction on the Internet. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020) (2020)
13. Nikolov, A., Radivchev, V.: Nikolov-radivchev at semeval-2019 task 6: Offensive tweet classification with bert and ensembles. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp. 691–695 (2019)
14. Parikh, P., Abburi, H., Badjatiya, P., Krishnan, R., Chhaya, N., Gupta, M., Varma, V.: Multi-label categorization of accounts of sexism using a neural framework. arXiv preprint arXiv:1910.04602 (2019)

15. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
16. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018)
17. Sadeque, F., Xu, D., Bethard, S.: Measuring the latency of depression detection in social media. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. pp. 495–503 (2018)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)