uOttawa at eRisk 2021: Automatic Filling of the Beck's Depression Inventory Questionnaire using Deep Learning

Diana Inkpen, Ruba Skaik, Prasadith Buddhitha, Dimo Angelov and Maxwell Thomas Fredenburgh

University of Ottawa, School of Electrical Engineering and Computer Science, 800 King Edward Avenue, Ottawa, ON, K1N 6N5, Canada

Abstract

This paper describes the University of Ottawa's participation in Task 3 of the eRisk 2021 shared task at CLEF 2021. We think that this task is important because it allows detecting the level of depression for social media users as often as needed, without having to ask them to spend their time manually filling in the Beck's Depression Inventory questionnaire. Our methods focus on selecting the relevant posts for each question of the questionnaire and using pre-trained deep learning models with or without fine-tuning to make predictions for unseen users.

Keywords

depression detection, social media, deep learning, natural language processing

1. Introduction

This paper described the uOttawa team's participation in Task 3 of the eRisk 2021 [1] shared task. The task is a continuation of Task 3 at eRisk 2019 and Task 2 at eRisk 2020, and the goal is to automatically estimate the level of depression from a user's social media postings.

We believe that this task can be very useful for monitoring users in special situations, without having to ask them to manually provide information. For example, a psychologist could post-monitor their patients after their recovery, with their consent. Another example is for people who need to spend long periods in conditions of isolation, such as arctic researchers or astronauts during long space flights (in the future).

We employed deep learning techniques to classify information extracted from the postings. Our focus was on selecting relevant posts for each type of information. Then we employed zero-shot learning via pre-trained models or we trained models based on sequence-to-sequence models for Question Answering (QA).

The rest of the paper is organized as follows. Section 2 gives more details about the task and shows statistics about the class distribution in the training data. Section 3 describes the

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

diana.inkpen@uottawa.ca (D. Inkpen); rskai034@uottawa.ca (R. Skaik); pkiri056@uottawa.ca (P. Buddhitha); dange042@uottawa.ca (D. Angelov); mfred075@uottawa.ca (M. T. Fredenburgh)
https://www.site.uottawa.ca/~diana/ (D. Inkpen)
0000-0002-0202-2444 (D. Inkpen); 0000-0002-8929-8602 (R. Skaik)

CEUR Workshop Proceedings (CEUR-WS.org)

methods that we used. Section 4 presents their results, and discusses them. Section 5 concludes and suggests directions of future work.

2. Task description

For each user, the participants in the shared task[1] were given the postings of that user for a certain time period. The task was to automatically fill in a standard depression questionnaire: the Beck's Depression Inventory (BDI). The questionnaire has 21 questions which assess the presence of feelings like sadness, pessimism, loss of energy, etc. Each question has 4 answers (0,1,2,3). Therefore the task becomes a classification task into 4 classes, for each question. The answers are targeting changes in a user's life.

There is variation in the expected answers. Two of the questions have seven answers instead of four, namely: 0, 1a, 1b, 2a, 2b, 3a, 3b. Therefore these two classifiers will need to classify into 7 classes. The two questions are question 16 (about sleep patterns) and question 18 (about appetite).

The training dataset provided by the task organizers for Task 3 is composed of 43,514 Reddit posts and comments written by 90 users who have answered 21-questions of the BDI questionnaire during the past two years. The test dataset consists of 19,803 posts and comments written by 80 users.pub. More details about the construction of the dataset are available in [2].

In table 1, we show statistics about the class distribution in the training data. For the two questions with 7 answers, the answers 1a, 1b were considered equivalent when performing these counts. They were counted as class 1. We did the same for for classes 2a, 2b and 3a, 3b, because they contribute the same number of points when assessing the depression level of a user. We can see for the statistics that class 0 is the biggest (35%), while class 1 is not far behind (33%). Choosing class 0 as the default class would therefore not necessary be a great choice for the classifiers. Class 2 is 19% and class 3 is the smallest (13%). If we look at the depression levels, 15% of the users have minimal depression, 30% mild depression, 24% moderate depression, and 30% severe depression. So, we can say that the data is not very imbalanced.

The evaluation measures used in the shared task are: the Average Hit Rate (AHR), the Average Closeness Rate (ACR), the Average Difference between Overall Depression Levels (ADODL), and the Depression Category Hit Rate (DCHR).

The Average Hit Rate (AHR) is the Hit Rate averaged over all the users. It is a strict measure that computes the ratio of cases where the automatically filled questionnaire has exactly the same answer as the real questionnaire.

The Average Closeness Rate (ACR) is the Closeness Rate averaged over all users. The Closeness Rate measure takes into account that the answers of the depression questionnaire represent an ordinal scale, and not only separate options. For example, if the user answered "0", a system whose answer is "3" should be penalised more than a system whose answer is "1".

The Average Difference between Overall Depression Levels (ADODL) measures the overall depression level estimated taking all responses as a sum of all the answers, looking for the depression level as a whole instead of some differences on each questionnaire answer prediction. It computes the difference between the overall depression level for the real and automated

[1] https://erisk.irlab.org/

Table 1
Statistics about the class distribution in the training data

```
90 users
Question 1 : 0 : 27 (30%) 1 : 47 (52%) 2 : 11 (12%) 3 : 5 (5%)
Question 2 : 0 : 22 (24%) 1 : 34 (37%) 2 : 20 (22%) 3 : 14 (15%)
Question 3 : 0 : 22 (24%) 1 : 35 (38%) 2 : 18 (20%) 3 : 15 (16%)
Question 4 : 0 : 28 (31%) 1 : 33 (36%) 2 : 23 (25%) 3 : 6 (6%)
Question 5 : 0 : 34 (37%) 1 : 32 (35%) 2 : 12 (13%) 3 : 12 (13%)
Question 6 : 0 : 60 (66%) 1 : 13 (14%) 2 : 11 (12%) 3 : 6 (6%)
Question 7 : 0 : 28 (31%) 1 : 17 (18%) 2 : 23 (25%) 3 : 22 (24%)
Question 8 : 0 : 28 (31%) 1 : 27 (30%) 2 : 23 (25%) 3 : 12 (13%)
Question 9 : 0 : 41 (45%) 1 : 37 (41%) 2 :  7 (7%)  3 : 5 (5%)
Question 10 : 0 : 42 (46%) 1 : 23 (25%) 2 : 8 (8%) 3 : 17 (18%)
Question 11 : 0 : 37 (41%) 1 : 31 (34%) 2 : 14 (15%) 3 : 8 (8%)
Question 12 : 0 : 28 (31%) 1 : 32 (35%) 2 : 8 (8%) 3 : 22 (24%)
Question 13 : 0 : 38 (42%) 1 : 21 (23%) 2 : 16 (17%) 3 : 15 (16%)
Question 14 : 0 : 38 (42%) 1 : 21 (23%) 2 : 20 (22%) 3 : 11 (12%)
Question 15 : 0 : 17 (18%) 1 : 32 (35%) 2 : 28 (31%) 3 : 13 (14%)
Question 16 : 0 : 17 (18%) 1 : 36 (40%) 2 : 24 (26%) 3 : 13 (14%)
Question 17 : 0 : 38 (42%) 1 : 31 (34%) 2 : 16 (17%) 3 : 5 (5%)
Question 18 : 0 : 32 (35%) 1 : 30 (33%) 2 : 15 (16%) 3 : 13 (14%)
Question 19 : 0 : 29 (32%) 1 : 25 (27%) 2 : 25 (27%) 3 : 11 (12%)
Question 20 : 0 : 21 (23%) 1 : 34 (37%) 2 : 21 (23%) 3 : 14 (15%)
Question 21 : 0 : 51 (56%) 1 : 18 (20%) 2 : 11 (12%) 3 : 10 (11%)

Total:  0 : 678 (35%) 1 : 609 (32%) 2 : 354 (19%) 3 : 249 (13%)

Minimal depression 14 (15.5%)
Mild depression 27 (30%)
Moderate depression 22 (24.5%)
Severe depression 27 (30%)
```

questionnaire. Then, the absolute difference (ad) between the real and the automated score is computed. Depression levels are integers between 0 and 63. The measure is normalised to be between 0 and 1 with the formula (63 - ad)/63. Then the average over all the users is computed.

The Depression Category Hit Rate (DCHR) measures the correctness of the estimation achieved over all users according to the well-established depression categories in psychology, with the following four categories of depression:

```
minimal depression (depression levels 0-9)
mild depression (depression levels 10-18)
moderate depression (depression levels 19-29)
severe depression (depression levels 30-63)
```

See the task overview paper for more details about the evaluation measures [1].

3. Methods

Our methods contained three steps: pre-processing the data, selecting relevant posts, and the classification to predict the answers for each question, based on zero-short learning or supervised learning.

3.1. Preprocessing

Each post was preprocessed as follows: the title of the post and the post text were concatenated. The contractions were expanded, and words between brackets were removed. Then punctuation and special characters were cleaned, and all the text was lowercased. The posts related to the forum monitoring (namely posts that notified users that they "broke the rules") were removed. Finally, all posts that had less than four characters were removed.

3.2. Selecting posts

For each question, we focused on selecting a subset of posts for each user. The goal was to keep only posts that are relevant to each question, in order to increase the probability of finding an answer about the topic of the question. We call this process filtering of the posts. We also experimented with keeping all posts, with the caveat that training deep learning models on long texts (the concatenation of all posts) is slow or sometimes problematic for BERT-like models.

We employed two methods for filtering posts. The first method is similarity-based. The similarity-based method utilized pre-trained sentence transformer models based on BERT or RoBERTa to embed each post and all the BDI answers. Then, the relatedness of each post to each answer of BDI answers is measured by computing the cosine distance between the post_embedding (*p*) and the answer_embedding (*a*), as shown in the Equation 1[2].

$$1 - \frac{p \cdot a}{\|p\|_2 \cdot \|a\|_2} \text{ where } \|x\|_2 \text{ is the 2-norm of x} \tag{1}$$

If the similarity value of a post with all questionnaires' answers is less than θ_1, then the post is excluded, since it means that the post is not related to any of the BDI questionnaire questions. In addition, if the difference between the maximum similarity and the minimum similarity is less than θ_2, then the post is excluded because it is considered a general post and not assisting in answering any of the BDI questionnaire questions in specific. It should be noted that not all the categories are discussed in the posts. Some categories appear more often than others. If there are very limited posts for a user, we consider all the posts of that user for the learning process (all the posts are included in the top *n* posts). Table 1 shows the number of posts for each BDI question as per RoBERTa similarity with θ_1= 0.6. It shows that the posts related to eating and sleeping habits are rare, and posts about guilt and punishment feelings are the most frequent.

The second method is topic-based. We leveraged topic modeling to help identify relevant posts for each question. We used `top2vec` [3] to divide all posts into topics that were then used to find relevant posts for each question. The `top2vec` algorithm automatically finds the

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html

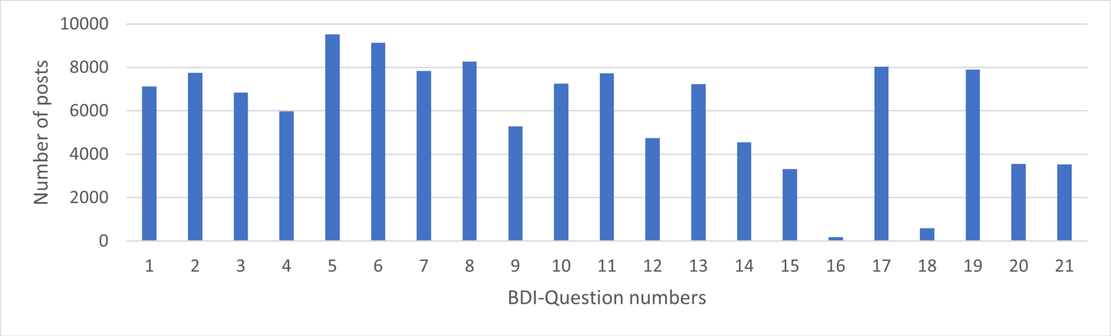


Figure 1: Number of posts per BDI-question based on RoBERTa (θ_1= 0.6)

number of topics in a corpus. It finds topic vectors from jointly embedded document and word vectors of a corpus. The main idea behind the algorithm is that it finds dense areas of documents in the embedding space. The assumption of the algorithm is that the dense area of document vectors represents an area of highly similar documents which are representative of a topic. A topic vector is calculated from each dense area of documents as the centroid of those document vectors. The topics are then described with the nearest word vectors to the topic vector. At the end, each document is assigned to its nearest topic vector, allowing for the size of each topic to be calculated. As an example, for the first question, we searched for the topics related to 'sad' and 'feel', using two relevant topics computed by `top2vec`, containing the following words:

cry, crying, sad, cried, feeling, upset, emotional, . . .

smile, eyes, myself, cry, face, laugh, beautiful, sad, beauty, wish, forgive, . . .

3.3. Classification

3.3.1. Zero-Shot Learning

The current dataset set is relatively small for training 21 classifiers for each question of the BDI questionnaire, especially if we considered only the posts that are related to the BDI question. In addition, it is difficult to relate which posts are answering which question. For that reason, we decided to utilize transfer learning, more precisely zero-shot learning. Zero shot learning is a fast emerging field in machine learning, with a broad range of computer vision and natural language processing applications [4].

We used a simple technique that relies on estimating the semantic similarity between the BDI answer and each post of the user. We did make use of the training labels to choose the thresholds θ_1 and θ_2. Therefore the method is not totally unsupervised. Fixed thresholds could be chosen if preferred.

We used language models based on Sentence Transformers for deep contextual post representations: Sentence-BERT (SBERT) and Sentence-RoBERTa (SRoBERTa) [5]. SBERT/SRoBERTa are modifications of the pretrained BERT/RoBERTa network that employs siamese and triplet network architectures [6] as illustrated in Figure 2.

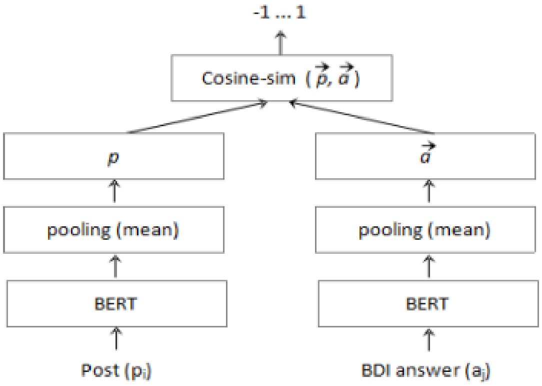


Figure 2: SBERT architecture. (https://www.sbert.net/).

The following runs from table 2 are based on zero-shot learning: uOttawa1_sim_BERT_base+, uOttawa3_sim_BERT_large and uOttawa5_sim_RoBERTa+. uOttawa1_sim_BERT_base+ uses 'bert-base-nli-mean-tokens' model. The model starts by getting the word embeddings for each word in a given sentence using BERT base model, then calculates the average of the word embeddings to produce SBERT embeddings. The weights are updated using siamese and triplet networks to construct semantically relevant sentence embeddings that can be compared using cosine-similarity. The base model contains 12-layers, 768-hidden layers, 12-heads, 110M parameters [7]. Whereas, uOttawa3_sim_BERT_large uses the large BERT model as the initial word embedding for each word in the sentence. It contains 24-layers, 1024-hidden layers, 16-heads, 340M parameters [7]. Similarly, uOttawa5_sim_RoBERTa+ is based on 'roberta-base-nli-mean-tokens' model.

3.3.2. BERT QA

One of the key limitations that BERT-based models [7] face is the length of the input sequence. Due to the full attention mechanism, the computational memory requirement and the model training time is quadratic. In other words, if the sequence length is n, the memory requirement when training the model will be n^2. Due to this reason, the authors of transformers architecture [8] and similar architectures, such as BERT, have limited the sequence length to 512 tokens, which includes the special tokens [CLS] (classification embedding) and [SEP] (sentence separator). To overcome this sequence length limitation, many researchers have introduced different architectures such as Longformer [9], Reformer [10] and BigBird [11]. Even though we conducted several experiments in the form of multiple-choice question answering using the Longformer architecture, we still found it resource-intensive, especially when the sequence length increases. To obtain the results for the "uOttawa4_Ensemble_BERT_QA" run (see 2) and "uOttawa6" (see table4), we conducted several preliminary experiments using the BigBird architecture in the form of multiple-choice question answering. For the experiments in this section, we did not filter the posts as mentioned in section 3.2 and tried to identify the impact of using the content published by users soon before submitting the BDI questionnaire (i.e., results under "uOttawa4_Ensemble_BERT_QA"). In addition, we also conducted several experiments by using all the earliest Reddit posts from the collection of posts of users (for the "uOttawa6" run). After pre-processing the data, we concatenated all the posts in the order of posts' date and time. The concatenated posts were then tokenized using the spaCy tokenizer, so that a limited number of tokens can be extracted for training purposes. Due to the resource intensiveness of the transformer-based models, we selected only 512 tokens from each user. It is important to note that these 512 tokens do not reflect the number of tokens that will get generated when using the BigBirdTokenizer [12] to tokenize the input data. We prepared our input, per user, per question, in the following format, which was then tokenized using the BigBirdTokenizer.

$$[[[CLS]\ T_1\ T_2\ T_3\ \ldots\ T_{510}\ T_{511}\ T_{512}\ [SEP]\ Q_1\ C_1^{(q_1)}\ [SEP]]\ \ldots$$
$$[[CLS]\ T_1\ T_2\ T_3\ \ldots\ T_{510}\ T_{511}\ T_{512}\ [SEP]\ Q_1\ C_4^{(q_1)}\ [SEP]]]$$

Here T_1 to T_{512} indicates the sequence of tokens (512 for our experiments) extracted from the concatenated posts of a user. Q_1 states the question, which will be from 1-21 that also indicates the number of classifiers trained. $C_1^{(q_1)}$ indicates the first choice of the first question, where the number of choices can vary between questions. According to the BDI questionnaire, except for questions 16 and 18, the questions had four choices, while questions 16 and 18 had seven choices. Based on the number of choices, we changed the number of class labels accordingly. We used the huggingface implementation of the BigBird model [12] to train the classifiers. Given the "content", "question" and the "choice" as mentioned above, we used the pre-trained BigBird model and fine-tuned it on our task as a multi-class classifier predicting one out of four or seven choices, based on the question. Unlike the BERT model, which uses full attention, the BigBird model uses random, window, and global attention to reduce the quadratic impact on training time and computational memory. When training, we used the Adam optimizer and trained the model for ten epochs with early stopping. We created five stratified shuffle splits [13] by allocating 80% for training and 20% for validation. For each stratified split, we created separate models and saved the ones that produced the best results on the validation data. During inference, the saved models were used on the test data to generate predictions, and the softmax outputs were aggregated to create an ensembled output. The stratification was based on the level of depression (i.e., minimal, mild, moderate, and severe depression) calculated based on the answer provided for each question.

It is important to note that even though the level of depression was used as the stratification strategy, the answers provided by the participants within the same depression group were not consistent. Due to resource intensiveness and training time, we trained the model with a batch size of 1. Given the time constraints, we fine-tuned only a few of the hyperparameters of the BigBird model. We set the block size of the model to be 64 and the number of random blocks to be 5. The block size specifies the block size to be used with random, global and window attention, and the number of random blocks specifies how many random blocks to be used with the given block size. We could not use a sequence length larger or equal to 1024 due to computational limitations. Though using sparse attention could be more effective if used with sequences longer than 1024 tokens [12]. Given these constraints, we will conduct further research in future work to identify more optimal hyperparameters to obtain better results.

3.3.3. Universal Sentence Encoder QA

Each user has unique Reddit posts which may be on a variety of different topics. In order to train a model to predict the answer of a user for a given BDI question, we would ideally only use the posts that are relevant to answering the question. However given the limited size of the dataset it would be difficult to train a model to learn which posts are relevant to a question and the user's response. There is the additional challenge that some users may have a very large quantity of posts that may not be able to be processed all at once by deep learning models due to computational constraints.

In order to overcome these challenges, we leverage transfer learning by using the universal sentence encoder [14] optimized for question answering (USE-QA). We use it to find the most semantically-relevant posts of a user for each BDI question. We accomplish this by using the response and question encoders of the USE-QA. We encode each of a user's posts with the response encoder. Then for each question, we create a question. For example for question 1, which is about sadness, we create "How sad do I feel?", for question 2, which is about pessimism, we create "How discouraged do I feel?". We then embed each one of these questions with the USE-QA question encoder. The question and response embeddings can then be compared using cosine similarity. In order to select the most relevant posts of a user to each question, we took their top 10 most similar posts to each question. Once we have identified the most relevant posts of a user to each BDI question we concatenate them together as they will be used to train a neural network for each BDI question which predicts the user's responses.

The neural architecture we use is motivated by the deep averaging network [15] and the need for transfer learning due to the small data size. For the embedding layer, we use USE to embed the concatenated top 10 most relevant posts for a BDI question. This is followed by three fully-connected layers with dropout and a final dense layer of size equivalent to the number of responses for the given BDI question. We use a 90%/%10 training and validation split on the provided training data (90 users) and train the models for 10 epochs. At prediction time for the test data (80 users) the USE-QA method was used to find the top 10 most relevant posts for each BDI question for each user, then those were concatenated and put through the trained neural network to predict user responses.

3.3.4. Hierarchical Attention Network

For this method, we also trained 21 classifiers for each of the BDI questions, but we adopted a hierarchical attention network (HAN) for document classification inspired by [16]. It employs two levels of attention mechanisms at the word and sentence levels as described in Appendix A. A word attention mechanism is utilized to identify keywords then aggregate them to create a sentence vector. Then a sentence attention mechanism is used to emphasize the importance of a sentence.

4. Results and Discussion

Table 2 shows the results of the 5 runs we submitted to the shared task, also available in the task overview paper [1]. Here is a description of the methods we used to obtained the predictions

Table 2
Results for the submitted runs

Run	AHR	ACR	ADODL	DCHR
uOttawa1_sim_BERT_base+	**28.39**%	**65.73**%	**78.91**%	25.00%
uOttawa2_top2vec_USE+	28.04%	63.00%	77.32%	27.50%
uOttawa3_sim_BERT_large+	25.83%	59.68%	71.23%	27.50%
uOttawa4_Ensemble_BERT_QA	27.68%	62.08%	76.92%	20.00%
uOttawa5_sim_RoBERTa+	26.31%	62.60%	76.45%	**30.00**%

for the test users.

uOttawa1_sim_BERT_base+

This run used SBERT, which is pre-trained on a natural language inference (NLI) dataset in addition to BERT's Wikipedia pre-training. As described in section 3.3.1, after calculating the cosine-similarity of each post against the BDI questionnaire answers, we filtered the unrelated and general posts based on θ_1 and θ_2. In this run, we kept all the post, thus we set $\theta_1 = 0$ and we removed general posts by setting $\theta_2 = 0.1$. Then, each post was assigned to the maximum similarity value of the BDI answer, as illustrated in table 3. Finally, the answers for each user were aggregated using voting for the most frequent answer, for each question of the BDI questionnaire. This is our best submitted run in terms of performance based on the first three evaluation measures.

Table 3
Cosine-similarity with the post *"this is so sad i cry every time"*

0	i do not feel sad	0.0902
1	i feel sad much of the time	0.8822
2	i am sad all the time	**0.9353**
3	i am so sad or unhappy that i can't stand it	0.9119

uOttawa2_top2vec_USE+

This used the method described in section 3.3.3, based on the Universal Sentence Encoder with a QA training architecture. Note that `top2vec` was not used in this method (we put the top2vec in the run's name by mistake).

uOttawa3_sim_BERT_large+

This run is based on zero-shot learning using the model 'bert-large-nli-stsb-mean-tokens'. This model is considered suitable for semantic textual similarity as it was fine-tuned on the NLI dataset, then on the sentence similarity STS benchmark train set. In this run, we kept all the posts as the uOttawa1_sim_BERT_base+ run, thus we set $\theta_1 = 0$ but we changed θ_2 to 0.5 to eliminate all relatively ambiguous posts.

Table 4
Results for unofficial runs

Run	AHR	ACR	ADODL	DCHR
uOttawa6	29.46%	63.04%	**78.31**%	**25.00**%
uOttawa7_Sim_HAN_cce_top_20_20	**32.62**%	**65.99**%	77.62%	22.50%
uOttawa8_Sim_HAN_cce_top_10	30.48%	63.63%	72.88%	22.50%
uOttawa9_Sim_HAN_cce	31.73%	64.62%	75.22%	**25.00**%

uOttawa4_Ensemble_BERT_QA

This run is based on a BigBird model using 512 tokens from the end of the concatenated and tokenized Reddit posts using the spaCy tokenizer (i.e., before tokenizing the sequence using the BigBirdTokenizer). The training uses the QA models described in section 3.3.2.

uOttawa5_sim_RoBERTa+

This run is based on zero-shot learning (section 3.3.1) with the use of pre-trained 'roberta-base-nli-mean-tokens' model. In this run, we set $\theta_1 = 0.25$, $\theta_2 = 0$, then we performed extra filtering of the posts, by removing any post with a maximum similarity to the BDI answer that is less than 0.6. This is our best submitted run in terms of the forth evaluation measure, the level of depression.

Table 4 shows results for four more runs, submitted unofficially since only 5 runs were allowed for the official submission. They were kindly evaluated by the task organizers. Here is the description of the methods used to produce these results.

uOttawa6

Then run "uOttawa6" is based on the architecture described in section 3.3.2.

uOttawa(7,8,9)_Sim_HAN_cce+

These three unofficial runs employed post filtering (based on similarity or on `top2vec`) and deep learning of the questionnaire answers based on hierarchical attention network. The posts filtering was done by either setting θ_1 to 0.5 and, if there are no posts that represent the category of the questionnaire, the posts filtered by `top2vec` were added. Or, by selecting the top n most-similar posts for each topic, then combining all the posts of the user as one document. We set n to 10 or 20. Table 5 shows the number of posts selected for each run.

We note that adding the supervised deep learning step (HAN), as described in section 3.3.4, helped improve the results (especially for the uOttawa7_Sim_HAN_cce_top_20_20 run). Table 7 form the Appendix B shows results for each question for three runs. The questions 16 and 18, about changes in eating and sleeping patterns, respectively, were the most difficult to answer.

Our unofficially submitted runs obtained better performance for the first measure (see tables 2 and 4), the correctness of the predicted answers (AHR 32.62% for uOttawa7_Sim_HAN_cce_top_20_20

Table 5
Number of posts included based on the selection criteria

run	n/θ_1	posts	train	test	post selection method
uOttawa7_Sim_HAN_cce_top_20_20	20	14727	9043	5776	RoBERTa Sim
uOttawa8_Sim_HAN_cce_top_10	10	9515	575	3798	RoBERTa Sim
uOttawa9_Sim_HAN_cce	0.5	6003	3570	2441	RoBERTa Sim & top2vec

versus 28.39% for uOttawa1_sim_BERT_base+), but not for the other measures. Table 6 compares our results with the best results from the shared task, for the four measures.

Table 6
Our results compared to the best results from the shared task

	Run	Rank	Our best	Best result
AHR	uOttawa7_Sim_HAN_cce_top_20_20	12	32.62%	35.36%
ACR	uOttawa7_Sim_HAN_cce_top_20_20	15	65.99%	73.17%
ADODL	uOttawa1_sim_BERT_base+	7	78.91%	83.59%
DCHR	uOttawa5_sim_RoBERTa+	5	30.00%	41.25%

5. Conclusion and Future Work

This paper presented several methods for the task of filling the BDI questionnaire. We showed that filtering posts by their relevance are beneficial in training classifiers related to answering different questions. We also showed that zero-shot learning with pre-trained models could be utilized for predicting the answers, with similar performance as QA sequence-to-sequence learning. In addition, deep learning models such as HAN on top of the post filtering led to our best results.

In future work, we plan to investigate possible ways to improve the performance. One direction is to further pre-train generic sentence similarity measures on large amounts of postings about mental health issues. Another direction is to investigate more ways to use `top2vec` to detect the most relevant posts for each question and to test better linguistic analysis methods for "change" detection to chose the correct answer to each question. Better ways to implement zero-shot learning can also be investigated.

Acknowledgments

We thank to the Natural Science and Engineering Council of Canada (NSERC) for supporting our research.

References

[1] J. Parapar, P. Martin-Rodilla, D. Losada, F. Crestani, Overview of eRisk 2021: Early Risk Prediction on the Internet, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Twelfth International Conference of the CLEF Association (CLEF 2021), Springer, 2021.

[2] D. E. Losada, F. Crestani, A Test Collection for Research on Depression and Language Use, Springer, 2016, pp. 28–39. doi:10.1007/978-3-319-44564-9_3.

[3] D. Angelov, Top2Vec: Distributed Representations of Topics, 2020. arXiv:2008.09470.

[4] W. Wang, V. W. Zheng, H. Yu, C. Miao, A survey of zero-shot learning: Settings, methods, and applications, ACM Trans. Intell. Syst. Technol. 10 (2019).

[5] 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.

[6] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815–823.

[7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[9] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The Long-Document Transformer, arXiv:2004.05150 (2020).

[10] N. Kitaev, L. Kaiser, A. Levskaya, Reformer: The Efficient Transformer, in: International Conference on Learning Representations, 2020. URL: https://openreview.net/forum?id=rkgNKkHtvB.

[11] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al., BigBird: Transformers for longer sequences, Advances in Neural Information Processing Systems 33 (2020).

[12] 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. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. 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.

[13] 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.

[14] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H.

Sung, et al., Multilingual universal sentence encoder for semantic retrieval, arXiv preprint arXiv:1907.04307 (2019).

[15] M. Iyyer, V. Manjunatha, J. Boyd-Graber, H. Daumé III, Deep Unordered Composition Rivals Syntactic Methods for Text Classification, in: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers), 2015, pp. 1681–1691.

[16] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 1480–1489.

A. Hierarchical Attention Network

This section describes the Hierarchical Attention Network (HAN) architecture we used for the multi-classification task for each category of the BDI questionnaire. The architecture is shown in figure 3. We trained 21 HAN-classifiers using the top-related posts for each category-based on the parameters described earlier. HAN employs bi-directional GRU on the word level, followed by an attention model, to extract the most informative words, which are then aggregated to generate a sentence vector, as shown in figure 4. Similarly, bi-directional LSTM on the sentence level is used with an attention mechanism to aggregate the most essential sentences to form the user-category vector which is then passed on to a dense layer for text classification using softmax activation as shown in figure 5 . For training, we use batch_size=128, Adam optimizer and categorical cross-entropy as the loss function. We added a dropout layer to avoid over-fitting.

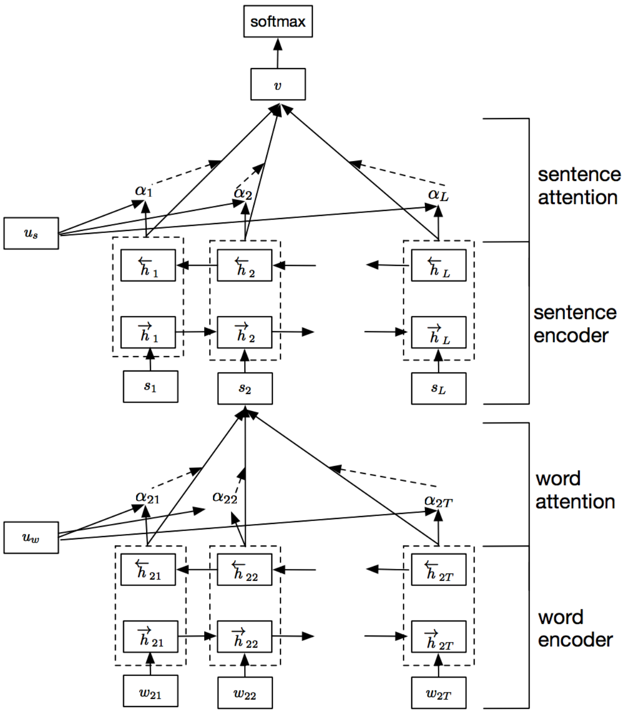
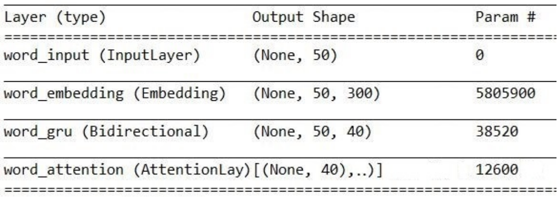


Figure 3: Hierarchical Attention Network [16].



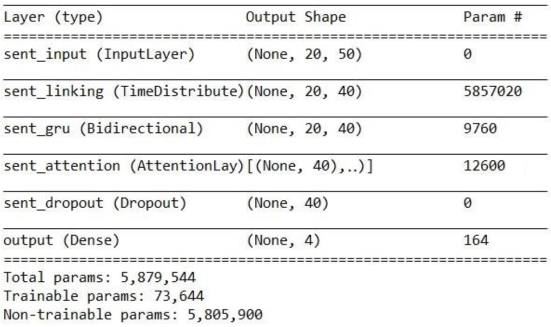
```
Layer (type)                 Output Shape              Param #
=================================================================
word_input (InputLayer)      (None, 50)                0

word_embedding (Embedding)   (None, 50, 300)           5805900

word_gru (Bidirectional)     (None, 50, 40)            38520

word_attention (AttentionLay)[(None, 40),..)]          12600
=================================================================
```

Figure 4: Word Encoder for Question 2 (Sim_HAN_cce_top_20_20).



```
Layer (type)                 Output Shape              Param #
=================================================================
sent_input (InputLayer)      (None, 20, 50)            0

sent_linking (TimeDistribute)(None, 20, 40)            5857020

sent_gru (Bidirectional)     (None, 20, 40)            9760

sent_attention (AttentionLay)[(None, 40),..)]          12600

sent_dropout (Dropout)       (None, 40)                0

output (Dense)               (None, 4)                 164
=================================================================
Total params: 5,879,544
Trainable params: 73,644
Non-trainable params: 5,805,900
```

Figure 5: Attention model summary for Question 2 (Sim_HAN_cce_top_20_20).

B. Results per Question

Table 7 shows the results for each question for three runs.

Table 7
AHR results per category for our best runs

Question	BDI Question	uOttawa1 AHR	uOttawa5 AHR	uOttawa7 AHR
1	Sadness	31.25	22.50	53.75
2	Pessimism	36.25	37.50	28.75
3	Past failure	32.50	36.25	38.75
4	Loss of pleasure	32.50	12.50	38.75
5	Guilty feelings	36.25	35.00	41.25
6	Punishment feelings	35.00	28.75	42.50
7	Self-dislike	17.50	32.50	25.00
8	Self-criticalness	22.50	21.25	28.75
9	Suicidal thoughts or wishes	46.25	38.75	36.25
10	Crying	32.50	31.25	32.50
11	Agitation	23.75	27.50	41.25
12	Loss of interest	32.50	26.25	27.50
13	Indecisiveness	31.25	22.50	23.75
14	Worthlessness	27.50	26.25	31.25
15	Loss of energy	23.75	25.00	20.00
16	Changes in sleeping pattern	12.50	17.50	23.75
17	Irritability	31.25	28.75	30.00
18	Changes in appetite	12.50	15.00	23.75
19	Concentration difficulty	27.50	32.50	22.50
20	Tiredness or fatigue	25.00	22.50	36.25
21	Loss of interest in sex	26.25	12.50	38.75