Measuring the Latency of Depression Detection in Social Media

Farig Sadeque, Dongfang Xu, Steven Bethard
School of Information, University of Arizona
1103 E 2nd St., Tucson, Arizona
{farig,dongfangxu9,bethard}@email.arizona.edu

ABSTRACT

Detecting depression is a key public health challenge, as almost 12% of all disabilities can be attributed to depression. Computational models for depression detection must prove not only that can they detect depression, but that they can do it early enough for an intervention to be plausible. However, current evaluations of depression detection are poor at measuring model latency. We identify several issues with the currently popular ERDE metric, and propose a latency-weighted F1 metric that addresses these concerns. We then apply this evaluation to several models from the recent eRisk 2017 shared task on depression detection, and show how our proposed measure can better capture system differences.

KEYWORDS

Social media; Depression; Latency; Neural networks.

ACM Reference Format:
Farig Sadeque, Dongfang Xu, Steven Bethard. 2018. Measuring the Latency of Depression Detection in Social Media. In *WSDM 2018: 11th Eleventh ACM International Conference on Web Search and Data Mining, February 5–9, 2018, Marina Del Rey, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3159652.3159725

1 INTRODUCTION

In their Global Burden of Disease 2000 study, the World Health Organization estimated that depression is responsible for more than four percent of the Disability-Adjusted Life Years (DALYs) lost and will be the second leading cause of DALYs lost, behind ischaemic heart disease, by 2020 if the trend continues [32]. Depression also accounts for 11.9% of all years Lived with Disabilities (YLDs) - the highest among all the mental and neurological conditions - with nearly 350 million people suffering from it worldwide [31]. In 2000, depression imposed an annual economic burden of 83 billion dollars in the US - most of which was attributed to reduced productivity and increased medical expenses [22]. Depression is also a major cause of suicide: according to a study by Goodwin and Jamison [10], 15-20% of all major depressive disorder patients take their lives. This outcome is largely avoidable if there are proper interventions, and early detection of depression is the first step towards these interventions. Most studies of early detection of depression rely on diagnoses based on patients' self-reported experiences and surveys [11]. The cost of these diagnoses is extremely high, and as of 2009, 30% of world governments who provide primary health care services do not have these programs [7].

The ubiquity of social media among the world population can provide a solution to this problem. Studies have shown associations between usage of social media and depression [23, 34]. Activities in social media can be used as predictors for well-being [21] and social participation [24]. But a key aspect of detecting depression in social media is the speed of detection: the longer we wait to intervene, the greater the risk of self harm. Hence, predicting depression early in a user's lifecycle is paramount. This argues for models that don't just look at one snapshot of a user's activities, but instead track the user's activities over time. It also argues for evaluation metrics that consider not only the precision and recall of detecting depressed users, but also the speed of that detection.

The recent eRisk 2017 shared task on depression detection in social media [16] introduced the early risk detection error (ERDE) metric to incorporate speed of detection into model evaluation. ERDE penalizes the score given to true positives (e.g., depressed users that the system also identified as depressed) based on how much of the data the system observed before making a prediction. However, ERDE has several drawbacks, including that 4 different meta-parameters, which have non-obvious consequences, must be defined before its use, and that the slow-prediction penalty under-penalizes fast systems while over-penalizing slow systems.

In this work, we make the following contributions:

(1) We introduce a new metric, latency-weighted F1 or F_{latency}, for measuring the quality and speed at which a model identifies whether a user is depressed given a series of their social media posts, and show how it addresses some of the drawbacks of ERDE.
(2) We propose a general approach for improving the latency of detection models based on checking the consistency of a model's predictions over a risk window.
(3) We evaluate several different models for depression detection in social media using both ERDE and our proposed F_{latency}, and show how the latter results in a more interpretable evaluation.

2 RELATED WORK

De Choudhury et al. [6] asked Amazon Mechanical Turk users to take the CES-D depression screening test and provide their Twitter handle. They then constructed support vector machine classifiers to distinguish between depressed and non-depressed users. Their models incorporated features such as posts per day, replies per day, shared interactions with other users, use of emotion words from the Linguistic Inquiry and Word Count (LIWC) lexicon, and use of a list of depression-related words mined from Yahoo! Answers

WSDM 2018, February 5–9, 2018, Marina Del Rey, CA, USA

ACM ISBN 978-1-4503-5581-0/18/02…$15.00
https://doi.org/10.1145/3159652.3159725

Mental Health. Their model achieved almost 70% accuracy, but was not evaluated for the speed at which it could make a prediction.

Wang et al. [30] studied images posted in Flickr[1] to identify self harm contents. They started with a set of posts tagged with *selfharm* and *selfinjury*, collected tags that occur more frequently with these two tags and then collected posts tagged with one or more of these tags. They only selected those users with more than five posts with these tags, and then manually examined whether the user is tagged correctly as someone with history of self harm. They used convolutional neural networks to classify images, and took advantage of the image titles for a better prediction result. Their best model achieved a 71 F1 score on the test set. This task was mostly inclined towards identifying users with self harm history, rather than users in risk of future self harm, and early detection was not an issue with the task.

For the 2016 CLPsych shared task [20], the mental health forum ReachOut annotated a set of posts with how urgently they needed moderator attention (red/amber/green). Systems competed to take a post of interest and the user's preceding history of posts, and classify the post of interest as red, amber or green. The most successful system in the shared task used various weightings of n-grams: Mac et al. [18] used TF-IDF weightings of unigrams along with post-level and sentence level embeddings using sent2vec [15], whereas Malmasi et al. [19] went through lexical features like n-grams ranging from 1 to 8 and syntactic features like parts of speech tags and dependencies. Both of these works implemented ensemble classification over sets of simpler classification models, and in both cases the ensemble model came out as the best. While this shared task considered prediction given a series of social media posts, it did not attempt to evaluate the speed of detection.

Observational latency has been occasionally considered in fields outside of social media analysis. For example, in the field of computer vision, observational latency has been used as a parameter to facilitate early detection of events [8, 12]. Hoai and De la Torre [12] used the number of frames a model requires to detect a facial expression as a parameter for the loss function of their prediction model. Ellis et al. went in a similar direction[8], using Microsoft Kinect data to detect human movement using the minimum number of frames possible. They showed how reducing the number of frames below a certain threshold can adversely affect the accuracy of the detection model.

The eRisk 2017 shared task [16] was the first work on depression detection in social media that focused on how quickly models could detect depressed users. The goal of the shared task was to identify depressed Reddit users as early as possible using the contents they have posted. The organizers searched for specific terms like "diagnosed with depression" to identify users who have been clinically diagnosed with depression at certain point, and collected all of these users' contents (up to 2000). They also collected a control dataset of random users who never mentioned a depression diagnosis in their posts. For each user (either depressed or control), their posts were divided into 10 chunks based on their posting time, where chunk 1 contained the earliest 10% of the user's posts and chunk 10 contained the last 10%. Submitted systems were evaluated on their correctness (precision, recall and f-measure) as well as their speed of detection (Early Risk Detection Error, ERDE, described in detail in Section 3.1). Submitted models used a wide range of feature extraction methods like n-grams, paragraph vectorization, external knowledge source incorporation etc. The best performing model used an ensemble over four word level n-gram logistic regression models, with higher F-measure than models that implemented more sophisticated techniques like recurrent neural networks.

The eRisk 2017 shared task was a great venue for encouraging the research community to focus on both the speed and quality of systems. However, as discussed in the next section, the ERDE metric under-penalizes fast systems, over-penalizes slow systems, and has a number of meta-parameters that are difficult to set. At the same time, the systems that competed in eRisk 2017 all used different model architectures and different feature sets, making it difficult to understand which components of each system were the biggest contributors to their performance. In the current paper, we address both of these issues, introducing a simpler and more interpretable measure for evaluating speed of detection, and examining various features and model architectures in comparable, side-by-side evaluations.

3 EVALUATION METRICS

3.1 Early Risk Detection Error (ERDE)

Early Risk Detection Error (ERDE) [17] aims at assessing models that take in a series of posts, and for each post, predict either '+' (depressed), '-' (not-depressed), or '?' (wait), where wait means the model wants to wait for more posts before making a prediction. Models are allowed to predict either '+' or '-' for a user only a single time, and as soon as one of these predictions is produced, the model is finished and will not be allowed to look at any more posts from that user. The goal of ERDE is to penalize models that abstain for a long time before predicting a user as depressed.

Formally, ERDE is defined as:

$$ERDE_o(U, sys) = \frac{1}{|U|} \sum_{u \in U} uERDE_o(u, sys)$$

$$uERDE_o(u, sys) = \begin{cases} c_{fp} & \text{if } ref(u)=- \wedge sys(u)=+ \\ c_{fn} & \text{if } ref(u)=+ \wedge sys(u)=- \\ c_{tp} \cdot lc_o(u, sys) & \text{if } ref(u)=+ \wedge sys(u)=+ \\ 0 & \text{if } ref(u)=- \wedge sys(u)=- \end{cases}$$

$$lc_o(u, sys) = 1 - \frac{1}{1 + e^{time(sys,u)-o}}$$

where U is the set of users, $ref(u)$ is the reference label ('+' or '-') assigned to the user, $sys(u)$ is the system's earliest non-'?' prediction, $time(sys, u)$ is the time (i.e., number of posts observed) for that earliest prediction, and where o, c_{fp}, c_{fn}, and c_{tp} are parameters of the model that must be set manually.

ERDE does penalize systems that take too long to make a prediction, but since it relies on a standard sigmoid centered at o, the transition between no penalty and 100% penalty is extreme. Figure 1 shows what the $ERDE_o$ penalty looks like for $o = 5$ and $o = 50$, the two values of o used in the eRisk 2017 evaluation. With $ERDE_5$, even a perfect system that correctly classified every user after only a single post would be penalized, since $1 - \frac{1}{1+e^{1-5}} > 0$. With $ERDE_{50}$, there is essentially no penalty for a system that takes 45 posts to

[1] http://www.flickr.com

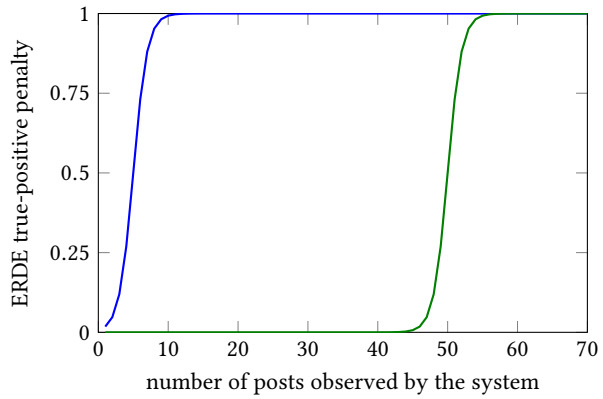

Figure 1: Plot of how the ERDE true-positive penalty increases with the number of posts observed by the system, for $o = 5$ (blue) and $o = 50$ (green)

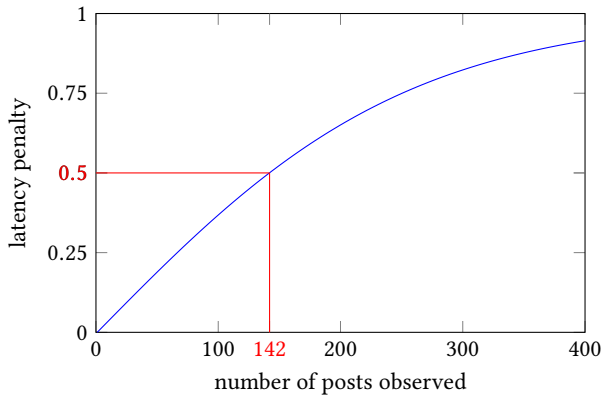

Figure 2: Plot of how the latency penalty increases with the number of posts observed by the system

predict depression, while a system that takes only 10 more posts to predict depression (55 posts) gets essentially no credit at all. We argue that such behavior is undesirable for a measure of speed of detection when, as was the case for eRisk 2017, there is no clear answer to the question "how many posts *should* it take to detect depression?"

ERDE has several additional drawbacks. Beyond the o parameter that we have just discussed, ERDE has 3 other parameters that must be manually set. In eRisk 2017, the organizers defined $c_{fn} = 1, c_{fp} = 0.1296$, and $c_{tp} = 1$, but these values were set heuristically, and it is not clear whether such values are appropriate or meaningful for other types of early detection tasks. ERDE is also not easily interpretable. The top system in eRisk 2017 achieved $ERDE_5 = 12.70\%$. Is that fast or slow? How many posts should one expect such a system to take to predict depression? ERDE is unable to answer such questions.

3.2 Latency and Latency-weighted F1

As an alternative to ERDE, we propose a simple, interpretable way of measuring how long it takes a system to predict a depressed user. We define the *latency* of a system to be the median number of posts that the system observes before making a prediction on a depressed user. Formally:

$$latency(U, sys) = \underset{u \in U \wedge ref(u)=+}{\text{median}} time(sys, u)$$

where, as above, U is the set of users, $ref(u)$ is the reference label ('+' or '-') assigned to the user, and $time(sys, u)$ is the time in number of posts observed for the system's earliest non-'?' prediction. Latency directly answers our earlier question: how many posts should one expect system sys to take to predict depression?

Latency, a measure of speed, should be coupled with measures of accuracy, like precision and recall, to give a complete picture of a system's performance. To produce a single overall measure that combines latency and accuracy, we introduce another metric *latency-weighted F1*. We define latency-weighted F1, or F_{latency}, as the product of a model's F1-measure (the harmonic mean of precision and recall) and the median of a set of penalties in the range [0, 1), which are determined by the model's time to predict each user. The penalty is 0 if a prediction is made after exactly 1 post is observed, and approaches 1 as the number of observed posts increases. Formally, we define:

$$P_{\text{latency}}(u, sys) = -1 + \frac{2}{1 + e^{-p \cdot (time(u,sys)-1)}}$$

$$F_{\text{latency}}(U, sys) = F_1(U, sys) \cdot \left(1 - \underset{u \in U \wedge ref(u)=+}{\text{median}} P_{\text{latency}}(u, sys)\right)$$

where $F_1(U, sys)$ is the F-measure of the system, defined in the standard way:

$$precision(U, sys) = \frac{|u \in U : sys(u) = ref(u) = +|}{|u \in U : sys(u) = +|}$$

$$recall(U, sys) = \frac{|u \in U : sys(u) = ref(u) = +|}{|u \in U : ref(u) = +|}$$

$$F_1(U, sys) = \frac{2 \cdot precision(U, sys) \cdot recall(U, sys)}{precision(U, sys) + recall(U, sys)}$$

F_{latency} has a single parameter that must be set, p, which defines how quickly the penalty should increase. We suggest that p should be set such that the latency penalty is 0.5 (i.e., 50%) at the median number of posts of a user. With this approach, p can be determined by fitting the $P_{latency}$ curve to two points: (0, 1) and (0.5, median-posts). In the eRisk 2017 data, the median number of posts of a user is 142, and fitting the $P_{latency}$ curve to (0, 1) and (0.5, 142) results in $p = 0.0078$. Figure 2 shows a plot of the resulting penalty. We argue that the shape of this penalty curve is much more appropriate than ERDE for measuring the speed of depression detection: models that predict correctly on the first post are unpenalized, and the penalty gradually increases as the number of posts observed increases. (In the early part of this curve, each post after the first that the model observes applies roughly a 0.5% penalty to F-measure.)

We believe that latency and F_{latency} improve over ERDE by (1) being more interpretable, (2) having fewer parameters that must be manually tuned, and (3) using a penalty that gradually increases with the number of posts observed.

4 MODELS

Neither the models nor the model predictions from eRisk 2017 are freely available, so we re-implement some of the common approaches applied to that task. This allows us not only to evaluate the models with F_{latency}, but also to set up head-to-head comparisons of feature, model architectures etc. This was not possible in the shared task format because each system had its own feature set and its own model architecture, so comparing across participating systems could not reveal which sub-components were most crucial for success on the task.

The two most popular approaches to the eRisk 2017 task of predicting a user's depression status from their posts before time t were:

Non-sequential Treat each user as a bag of features, aggregated across all of the user's posts before time t. These bags of features were typically fed to linear classifiers such as a support vector machine or logistic regression [1, 25, 29].

Sequential Treat each user as a sequence of bags of features, where each bag of features corresponds to a single post by the user before time t. These sequences of bags of features were typically fed to sequential classifiers such as recurrent neural networks [9, 25, 29].

Some of the top models took advantage of ensemble approach, where results of various other classifiers were used as an input to an ensemble classifier [25, 29]. These ensembles provided better performance than the individual models.

Popular features for such models in eRisk 2017 included:

Words Words from the posts, sometimes also including bigrams and trigrams of words [1, 29].

Depression words Words commonly associated with depression, extracted from external resources such as medical webpages [25].

Medical concepts Words associated with medical symptoms, diseases, procedures, etc. extracted from external resources like the Unified Medical Language System (UMLS) or WebMD.[25, 29]

In the following sections, we describe our implementation of such models and features, following the descriptions of the task participants.

4.1 Non-sequential model

Our non-sequential model takes a feature vector summarizing all the posts of a user before time t and tries to predict whether the user is depressed or not. To aggregate post-level feature vectors into a user-level feature vector, raw counts are converted into proportions. For example, when using bag-of-words features, each feature represents the proportion of time that one word in the vocabulary appears the user's posts. We feed the user-level feature vectors to a support vector machine (SVM) classifier. Figure 3 shows the architecture of this model.

We use Weka's SVM implementation [33], with a polynomial kernel, probability estimate outputs enabled, attribute normalization enabled, complexity constant set to 1, tolerance parameter set to 0.001, and epsilon set to 1.0^{-12}. We explored a few other parameter values on the training data, but did not see any improvements.

Feature	Feature vector length
Words	9930
Depression words (DepWords)	200
Depression embeddings (DepEmbed)	64
Medical concepts (Metamap)	404

Table 1: Summary of the features

4.2 Sequential model

Our sequential model takes a sequence of feature vectors, each summarizing one of the posts of a user before time t, and tries to predict whether the user is depressed or not. We use a recurrent neural network model, in which post-level feature vectors are fed into a layer of Gated Recurrent Units (GRU) [4], the first GRU layer is followed by a second GRU layer, and the last output of the second GRU layer is fed through a sigmoid layer to produce a binary output. Figure 4 shows the architecture of this model.

We use the Keras implementation of GRUs, with 128 GRU units for all GRU layers. To avoid overfitting, dropout [26] with probability 0.3 was used on input to the the first GRU layer. We used RMSProp optimization [27] on mini-batches of size 200, and followed the standard recommendations to (1) tune the learning rate, which we set to 0.002 based on preliminary experiments, and (2) leave the other hyperparameter settings at their default values. Each model is trained for at most 800 epochs, with training time around two hours using two Graphics Processing Units (GPUs).

4.3 Features

The features we implemented, inspired by the participants in the eRisk 2017 task, are briefly summarized in Table 1 and described in detail below.

4.3.1 Word features. These are count-based features capturing the number of times that words appeared in the text. We first segmented each post into words using Stanford CoreNLP [28], and replaced all numbers, URLs, subreddits, and punctuation marks with placeholders. Then we gathered a vocabulary by analyzing the entire dataset, counting the occurrences of each unique word, and retaining as part of the vocabulary only words with frequencies equal or greater to the median, 47. We ended up with 9930 features: 9926 unique words and 4 identifiers for numbers, URLs, subreddits and punctuation marks.

4.3.2 Depression word features (DepWords). These are count-based features capturing the number of times that unigrams and bigrams commonly associated with depression, e.g., *depressed* or *anxiety*, or *my depression* or *panic attacks*, appeared in the text. We first collected two sets of texts, one representing language commonly used when talking about depression, and one representing more general language. For the depression language, we drew the most recent posts (not comments) in the "top" and "hot" section of the *depression* subreddit, resulting in 1987 posts. Posts in this subreddit are not necessarily posted by users who are depressed, but are generally topically related to depression. For the general language, we drew the most recent posts in the "top" and "hot" section of the *textventures* subreddit, resulting in 1082 posts. Posts

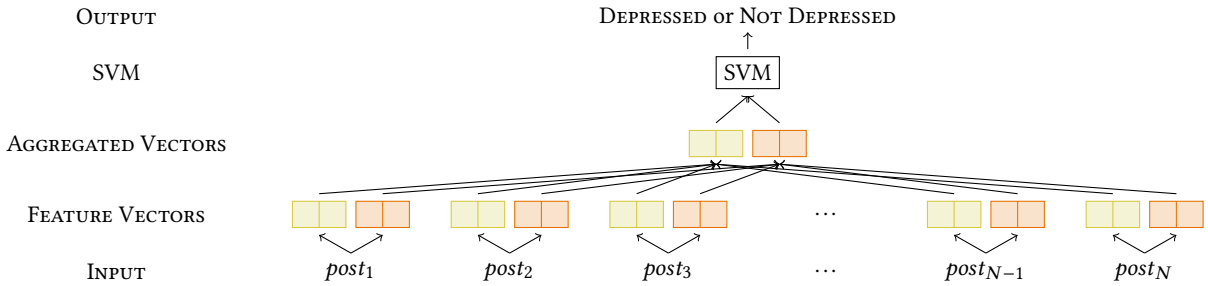

Figure 3: Architecture of the non-sequential model for predicting depression status from a user's posts.

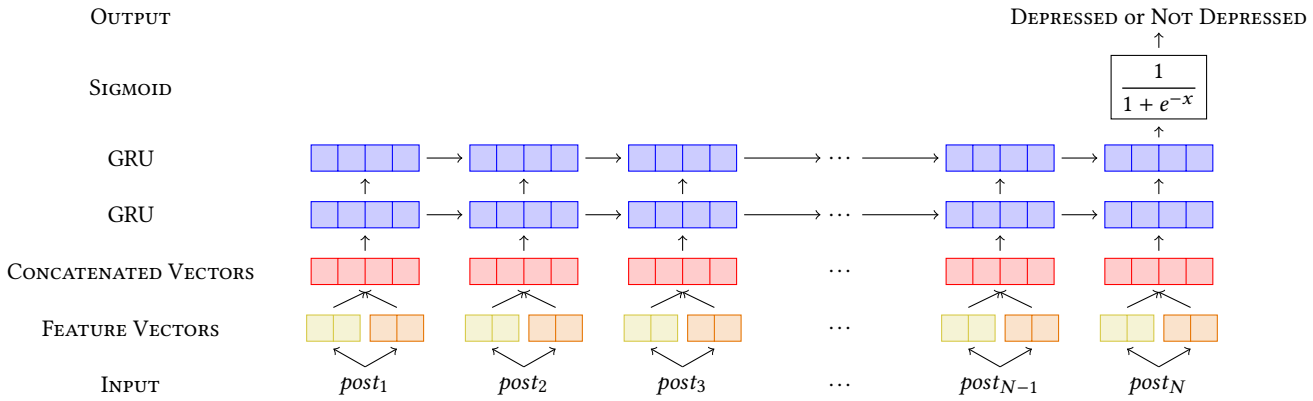

Figure 4: Architecture of the sequential model for for predicting depression status from a user's posts.

in this subreddit tell the beginning of a story (which commenters further develop), and cover a wide range of topics. We then used pointwise mutual information [5] to identify the top unigrams and bigrams most associated with the Depression subreddit. We ended up with 200 features: 100 top unigrams and 100 top bigrams.

4.3.3 Depression embedding features (DepEmbed). These are numeric features from a recurrent neural network that was trained to distinguish between depression-related language and other language. The network treats an entire post as a sequence so that it can capture linguistic phenomena that stretch over many words (e.g., *I just hit rock-bottom*), which cannot be captured by the previous features that treat a post as a bag of n-grams. We use a recurrent neural network in which the words in a post are fed into an embedding layer (128 dimensions), a Long Short-Term Memory (LSTM) [13] recurrent layer combines this sequence of embedded words into a dense vector (64 dimensions), and the result is fed through a sigmoid layer to produce a binary output. The architecture is shown in fig. 5. We train this model on the depression/textventures data from the DepWords features, asking the model to classify whether a post is from the depression subreddit or the textventures subreddit. We use an Adam optimizer [14] for training and dropout [26] with probability 0.15 to avoid overfitting. Once the model is trained, we discard the sigmoid layer, run the model on the posts from the eRisk 2017 data, and use the dense vector produced by the LSTM layer as the features. We thus ended up with 64 features: the 64 dimensions of the model's LSTM layer.

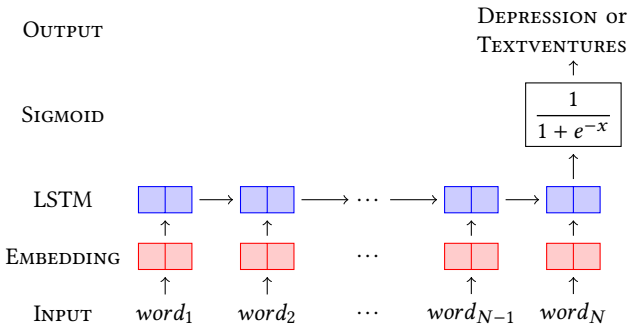

Figure 5: Architecture for training a model that can semantically summarize the contents of a post as a dense vector.

4.3.4 Medical concept features (Metamap). These are count-based features capturing the number of times concepts from the Unified Medical Language System (UMLS) [3], e.g., "Depressed mood" (C0344315), appeared in the text. We used the Metamap tool [2] to extract such concepts in the form of Concept Unique Identifiers (CUIs). We restricted Metamap to SNOMEDCT-US and to two semantic types: Mental or Behavioral Dysfunction and Clinical Drugs[2]. We passed each post through Metamap and counted the

[2]We included these restrictions because MetaMap produces a large number of spurious concept matches in social media data (e.g., the pronoun *I* is identified as Iodine; see

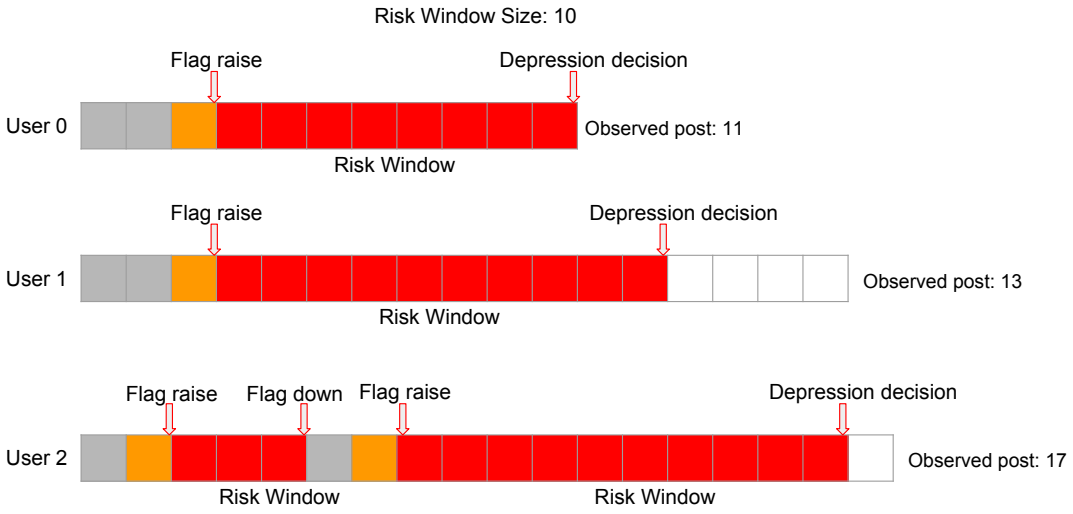

Figure 6: Example of post-by-post depression prediction with a risk window of size 10. Each block represents 1 post: gray is observed, orange is where the flag was raised, red is in the risk window, and white is unobserved. User 0 is an example where there are fewer remaining posts than the risk window, and user 2 is an example of restarting after a broken streak.

number of times each CUI occurred in the post. We ended up with 404 features: the 404 CUIs that MetaMap identified at least once in the eRisk 2017 training data.

5 PREDICTING WITH RISK WINDOWS

In preliminary analysis of the models above, we found it was common for a model to make occasional mistakes. But recall that in early depression prediction, the first "+" or "-" prediction is considered final, so occasional mistakes will force a early detection model to abort entirely, even if they have seen only a small number of posts so far. This can have a significant impact on their performance.

We thus introduce a technique, which can apply generally to any model, that trades off between latency and precision. If the model makes a prediction that the user is depressed after post p (we refer to this as *raising the flag*), we only confirm that prediction if the model continues to make the same (depressed) prediction for the next n posts (we refer to this as the *risk window*), or, if the user has fewer than n posts remaining, continues to make the same (depressed) prediction for all of their remaining posts. Figure 6 demonstrates the process with a risk window of size 10.

6 EXPERIMENTS

6.1 Data

All experiments were performed on the data published for the pilot task Early Detection of Depression in CLEF eRisk 2017 [16]. The organizers collected this data from Reddit[3], a social media and news aggregation website. Two sets of users were collected from Reddit: a depressed set and a non-depressed set. The depressed set was identified by searching the *depression* subreddit for a set of key phrases (e.g., *diagnosed with depression*), and then filtering the resulting users down to only those users for which the organizers could confirm, via a manual examination of the posts, that the user had explicitly declared that a physician had diagnosed them with depression. The non-depressed set included both random redditors from non-*depression* subreddits and random redditors who were active on the depression subreddit but had reported no depression[4]. For users from both sets, the organizers collected up to 2000 posts and comments per user, anonymized redditor ids, and then published the title, time and text of the posts. Their final collection contained 531,453 submissions from 892 unique users, but they released only part of this collection as a training dataset for the shared task; statistics for that subset are shown in Table 2.

	Train		Test	
	Depressed	Control	Depressed	Control
Subjects	83	403	52	349
Submissions	30,851	264,172	18706	217.665
Submissions/subject	371.7	655.5	359.7	623.7
Words/submission	27.6	21.3	26.9	22.5
Activity period (days)	573.23	627.17	608.8	623.7
Minutes between posts	920.99	994.08	1211.1	926.5

Table 2: Summary of the eRisk 2017 data

6.2 Model selection

Our two model architectures (non-sequential and sequential) and four feature sets (Words, DepWords, DepEmbed, and MetaMap) can be combined to create a large number of models. In this section, we use five-fold cross-validations on the training data to explore

also Appendix A). Moreover, we did not see any improvement using the 1000 most common CUIs from any semantic type in preliminary experiments.

[3]http://www.reddit.com

[4]The organizers acknowledged the possibility of having a non-depressed person in the depression group or vice versa, but suggested such events should be uncommon and the effects negligible, and pointed out that other identification strategies like questionnaires may suffer from this as well.

Model	Features	Precision	Recall	F_1
SVM	Words	53.3	38.6	44.8
SVM	DepWords	**77.3**	20.5	32.4
SVM	DepWords+Metamap	49.0	30.1	37.3
SVM	DepEmbed	69.4	30.1	42.0
SVM	DepEmbed+DepWords	66.7	45.8	54.3
SVM	DepEmbed+DepWords+Metamap	53.9	66.3	59.5
GRU	Words	72.8	51.8	60.6
GRU	DepWords	62.0	68.7	65.1
GRU	DepWords+Metamap	67.0	**75.9**	**71.2**
GRU	DepEmbed	65.8	60.2	62.9
GRU	DepEmbed+DepWords	60.0	61.4	60.7
GRU	DepEmbed+DepWords+Metamap	60.0	62.7	61.2

Table 3: Comparison of different models and feature sets in five-fold cross-validations on the training set when considering the entire posting history (window=∞).

which model/feature combinations look most promising, so that those can be evaluated on the test set. We focus for now on the simpler setup where the model observes a user's entire posting history (window=∞), and is evaluated just in terms of precision and recall.

Table 3 shows the cross-validation performance of a variety of models on the training data. The best F_1, 71.2, is achieved by the sequential (GRU) model with depression words (DepWords) and UMLS medical concept (MetaMap) features. Comparing across types of models, the sequential models are the clear winners: even the worst sequential (GRU) model had a higher F_1 than the best non-sequential (SVM) model (60.6 vs. 59.5). This finding is intuitive, given that early detection is a sequential prediction problem. Comparing across types of features, adding medical concepts (MetaMap) always improved F_1, but the results for other types of features were more mixed. Depression embeddings (DepEmbed) always improved the non-sequential (SVM) models, but always hurt the sequential (GRU) models. And using all words (Words) was better than just the depression words (DepWords) for the non-sequential (SVM) model, but the reverse was true for the sequential (GRU) model.

Looking across all the models, we selected two models for evaluation on the test set: the best non-sequential (SVM) model (DepEmbed+DepWords+ Metamap) and the best sequential (GRU) model (DepWords+Metamap). For each of these models, we apply a risk window as described in Section 5, considering all possible risk windows between 0 and the maximum number of posts, and optimizing the window size to maximize cross-validation F_{latency} on the training set. For the SVM model, an 11-post risk window yields the highest F_{latency} (67.1, with an F_1 of 82.0), while for the GRU model, a 23-post risk window yields the highest F_{latency} (52.6, with an F_1 of 65.7).

6.3 Evaluation

Table 4 evaluates the best models on the eRisk 2017 test set. For contrast, we also show each model with a risk window of 0 (i.e., the first '+' or '-' prediction is final) and a risk window of ∞ (i.e., the model always waits for all of a user's posts and decides at the final post).

Model	Risk window	$ERDE_5$	$ERDE_{50}$	F_1	Latency	F_{latency}
SVM	0	13.1	9.7	51.3	63.5	38.9
SVM	11 (best)	13.6	10.1	51.4	75	36.8
SVM	∞	13.2	11.7	45.4	199	16.0
GRU	0	12.5	9.4	33.5	9	32.3
GRU	23 (best)	15.2	11.5	44.4	69.5	32.7
GRU	∞	15.0	13.6	45.0	199	15.8

Table 4: Comparison of the top non-sequential and sequential models (SVM:DepEmbed+DepWords+Metamap and GRU:DepWords+Metamap) on the test set. For contrast, the same models are also shown with risk windows of 0 and ∞.

Comparing ERDE to F_{latency}, we see that F_{latency} is better at discriminating between models. For example, the non-sequential (SVM) and sequential (GRU) models with risk window 0 have given very similar values for ERDE, with their $ERDE_5$s differing by only 0.6 points and their $ERDE_{50}$s differing by only 0.3 points. Yet these two models have hugely different performance characteristics: the GRU is extremely fast (latency 9) at a significant cost to accuracy (F_1 of 33.5), while the SVM is much more cautious (latency 63.5) and much more accurate (F_1 of 51.3). Table 4 also shows the challenge of setting the ERDE o parameter: with $o = 5$ as in eRisk 2017, ERDE can't distinguish (only a 0.1 point difference) between a non-sequential (SVM) model that sees a median of 63.5 posts (window=0) and one that sees a median of 199 posts (window=∞), despite the latter being much, much slower to make predictions. We see these empirical results as a strong indication that F_{latency} better captures the important evaluation characteristics of early detection problems.

We found that the models with risk windows optimized on the training set (SVM:window=11 and GRU:window=23) did not always outperform other simple choices of risk window (window=0 or window=∞) on the test set. While the 23-window GRU model indeed outperformed the F_{latency} of the other GRUs (GRU:0 and GRU:∞), the 11-window SVM model did not have a better F_{latency} than the 0-window SVM; the tiny improvement in F_1 achieved by SVM:11 over SVM:0 was outweighed by its larger jump in latency.

Despite the training set results where sequential models substantially out-performed non-sequential models, on the test set the no-risk-window non-sequential (SVM) model outperformed all sequential (GRU) models, in terms of both F_{latency} and F_1. But note that on the training set, we compared systems with access to the entire posting history (window=∞), and, as can be seen in Table 4, the performance of the SVM model is much worse with such a large risk window. Probably the simple way that the non-sequential model aggregates feature vectors makes it easy to lose the signal of a single depressed post in a sea of many non-depressed posts.

7 LIMITATIONS

First, while it would have been ideal to compare ERDE to F_{latency} on the actual systems submitted to eRisk 2017, since neither code nor predictions were publicly available for any of the top systems, we had to approximate such systems by exploring combinations of the most common models and features in the task. We believe this still results in a nice contribution, as the models and features can be more

directly compared, but since our models are re-implementations, their performance on the task may differ somewhat from the systems submitted to the task.

Second, during model selection we first selected model architectures and feature sets under the maximal risk window, and then searched over all possible risk window sizes to select the best model in terms of F_{latency}. This two-stage procedure was not ideal for our non-sequential models, where it turned out that using no risk window, instead of the maximal one, resulted in the best models on the test set. A better approach would be to optimize risk windows simultaneously with feature sets and model architectures. There are also probably further gains to be had by directly optimizing for F_{latency} (e.g., instead of accuracy) during model training.

Third, F_{latency} combines F_1 and latency under the assumption that systems generally want to optimize F_1. However different applications may need to optimize different evaluation measures. For example, if the goal is to have a human intervene when a risk of depression is detected in a social media user, then probably a high recall even at the expense of precision would be preferred, so that the human would be able to intervene wherever possible. On the other hand, if the goal is to have an automatic intervention when a depression risk is detected, then probably a high precision is needed so that the automatic intervention is only applied when the model is very certain of the depression risk. Future work may need to extend F_{latency} to such scenarios, perhaps by including something like F_β's parameter for trading off between precision and recall.

Finally, F_{latency} is a general metric, applicable to any problem where systems must examine a sequence of items associated with an object, and make a prediction about that object's class as rapidly as possible. However, in the current paper, we only explore F_{latency} as applied to early detection of depression on social media. Future work will need to investigate the utility of F_{latency} on other kinds of problems: detecting drug discontinuation, churn prediction, etc.

8 CONCLUSION

We introduced latency and F_{latency} as evaluation metrics for early detection tasks, and showed that the theoretical behavior of these metrics is preferable to the current state-of-the-art, early risk detection error (ERDE). We replicated common models and features from the eRisk 2017 shared task on early detection of depression in social media, and showed empirically that our metrics are better than ERDE at capturing important differences between models. We also introduced the concept of risk windows for helping models find an acceptable trade-off between precision and latency, and showed that it successfully improves the performance of sequential prediction models on the eRisk 2017 test set. We believe our proposed metrics can be useful in the broad range of applications where models need to make fast user-level predictions from a sequence of the user's social media interactions.

ACKNOWLEDGEMENTS

This work was supported by National Institutes of Health grant R01GM114355 from the National Institute of General Medical Sciences (NIGMS). The computations were done in systems supported by the National Science Foundation under Grant No. 1228509. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health or National Science Foundation.

REFERENCES

[1] Hayda Almeida, Antoine Briand, and Marie-Jean Meurs. 2017. Detecting Early Risk of Depression from Social Media User-generated Content. In *8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017.*

[2] Alan R Aronson and Francois-Michel Lang. 2010. An overview of MetaMap: historical perspective and recent advances. In *Journal of the American Medical Informatics Association 17(3).* 229–236. https://doi.org/10.1136/jamia.2009.002733

[3] Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research* 32, suppl_1 (2004), D267–D270. https://doi.org/10.1093/nar/gkh061

[4] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8).*

[5] Kenneth Ward Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics* 16, 1 (1990), 22–29.

[6] Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting Depression via Social Media.. In *ICWSM.* 2.

[7] Roger Detels. 2009. *The Scope and Concerns of Public Health.* Oxford University Press Inc., New York.

[8] Chris Ellis, Syed Zain Masood, Marshall F. Tappen, Joseph J. Laviola, Jr., and Rahul Sukthankar. 2013. Exploring the Trade-off Between Accuracy and Observational Latency in Action Recognition. *Int. J. Comput. Vision* 101, 3 (Feb. 2013), 420–436. https://doi.org/10.1007/s11263-012-0550-7

[9] Marcelo L. Errecalde, Ma. Paula Villegas, Dario G. Funez, Ma. JosÃĺ Garciarena Uelay, and Leticia C. Cagnina. 2017. Temporal Variation of Terms as concept space for early risk prediction. In *8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017.*

[10] Frederick K. Goodwin and Kay Redfield Jamison. 1990. *Manic-Depressive Illness: Bipolar Disorder and Recurring Depression.* Oxford University Press Inc., New York.

[11] Aron Halfin. 2007. Depression: the benefits of early and appropriate treatment. *The American journal of managed care* 13, 4 Suppl (November 2007), S92âĂŤ7. http://europepmc.org/abstract/MED/18041868

[12] Minh Hoai and Fernando De la Torre. 2014. Max-Margin Early Event Detectors. *International Journal of Computer Vision* 107, 2 (01 Apr 2014), 191–202. https://doi.org/10.1007/s11263-013-0683-3

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. International Conference on Learning Representation.

[15] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *CoRR* abs/1405.4053 (2014). http://arxiv.org/abs/1405.4053

[16] David Losada, Fabio Crestani, and Javier Parapar. 2017. CLEF 2017 eRisk Overview: Early Risk Prediction on the Internet: Experimental Foundations. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum.*

[17] David E Losada and Fabio Crestani. 2016. A Test Collection for Research on Depression and Language Use. In *International Conference of the Cross-Language Evaluation Forum for European Languages.* Springer International Publishing, 28–39.

[18] Sunghwan Mac Kim, Yufei Wang, Stephen Wan, and Cecile Paris. 2016. Data61-CSIRO systems at the CLPsych 2016 Shared Task.. In *CLPsych@ HLT-NAACL.* 128–132.

[19] Shervin Malmasi, Marcos Zampieri, and Mark Dras. 2016. Predicting Post Severity in Mental Health Forums. In *The 3rd Workshop on Computational Linguistics and Clinical Psychology.* 133–137.

[20] David N. Milne, Glen Pink, Ben Hachey, and Rafael A. Calvo. 2016. CLPsych 2016 Shared Task: Triaging content in online peer-support forums. In *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology.* Association for Computational Linguistics, San Diego, CA, USA, 118–127. http://www.aclweb.org/anthology/W16-0312

[21] Michael J Paul and Mark Dredze. 2011. You are what you Tweet: Analyzing Twitter for public health. *Icwsm* 20 (2011), 265–272.

[22] Greenberg PE, Kessler RC, Birnbaum HG, Leong SA, Lowe SW, Berglund PA, and Corey-Lisle PK. 2003. The economic burden of depression in the United States: how did it change between 1990 and 2000?. In *J Clin Psychiatry 64(12).* 1465–1475.

[23] Brian A. Primack, Ariel Shensa, CÃĺsar G. Escobar-Viera, Erica L. Barrett, Jaime E. Sidani, Jason B. Colditz, and A. Everette James. 2017. Use of multiple social media platforms and symptoms of depression and anxiety: A nationally-representative study among U.S. young adults. *Computers in Human Behavior* 69 (2017), 1 – 9. https://doi.org/10.1016/j.chb.2016.11.013

[24] Farig Sadeque, Ted Pedersen, Thamar Solorio, Prasha Shrestha, Nicolas Rey-Villamizar, and Steven Bethard. 2016. Why do they leave: Modeling participation in online depression forums. In *Proceedings of the 4th Workshop on Natural Language Processing and Social Media*. 14–19.

[25] Farig Sadeque, Dongfang Xu, and Steven Bethard. [n. d.]. UArizona at the CLEF eRisk 2017 Pilot Task: Linear and Recurrent Models for Early Depression Detection. ([n. d.]).

[26] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[27] Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4, 2 (2012).

[28] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 173–180.

[29] Marcel Trotzek, Sven Koitka, and Christoph M. Friedrich. 2017. Linguistic Metadata Augmented Classifiers at the CLEF 2017 Task for Early Detection of Depression. In *8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017*.

[30] Yilin Wang, Jiliang Tang, Jundong Li, Baoxin Li, Yali Wan, Clayton Mellina, Neil O'Hare, and Yi Chang. 2017. Understanding and Discovering Deliberate Self-harm Content in Social Media. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 93–102.

[31] World Health Organization WHO. 2001. The world health report 2001- Mental Health: New Understanding, New Hope. http://www.who.int/whr/2001/en/whr01_en.pdf?ua=1. (2001). Last Accessed: 2016-04-02.

[32] World Health Organization WHO. 2003. Global Burden of Disease (GBD) 2000: version 3 estimates. http://www.who.int/entity/healthinfo/gbdwhoregionyld2000v3.xls?ua=1. (2003). Last Accessed: 2016-04-08.

[33] Ian H Witten and Eibe Frank. 1999. Data mining: practical machine learning tools and techniques with Java implementations. (1999).

[34] Liu yi Lin, Jaime E. Sidani, Ariel Shensa, Ana Radovic, Elizabeth Miller, Jason B. Colditz, Beth L. Hoffman, Leila M. Giles, and Brian A. Primack. 2016. Association between Social Media Use and Depression among U.S. Young Adults. In *Depression and Anxiety, 33(4)*. 323–331. https://doi.org/10.1002/da.22466

A APPENDIX: METAMAP

Metamap tries to identify UMLS concepts within a text, but has been designed primarily for biomedical use and not for social media. Below is an example of MetaMap applied to the following post from the depression subreddit:

> Nobody really gives a shit how depressed you are as long as you don't kill yourself, but if you did then they wonder why you didn't ask for help. This is a messed up world.

Running Metamap on this post will produce something like:

> Nobody really [gives: PREFERRED NAME='GIVE - DOSING INSTRUCTION IMPERATIVE', CUI='C1947971', SEMTYPES='[FTCN]', TRIGGER='["GIVE - DOSING INSTRUCTION IMPERATIVE"-TX-1-"GIVES"-VERB-0]'] a shit how [depressed: PREFERRED NAME='DEPRESSED MOOD', CUI='C0344315', SEMTYPES='[FNDG]', TRIGGER='["DEPRESSED MOOD"-TX-1-"DEPRESSED"-VERB-0]'] you are as [long: PREFERRED NAME='LONG', CUI='C0205166', SEMTYPES='[QLCO]', TRIGGER='["LONG"-TX-1-"LONG"-ADV-0]'] as you don't [kill: PREFERRED NAME='KILLING', CUI='C0162388', SEMTYPES='[SOCB]', TRIGGER='["KILLING"-TX-1-"KILL"-VERB-0]'] yourself, but if you did then they wonder why you didn't ask for [help: PREFERRED NAME='ASSISTED (QUALIFIER VALUE)', CUI= 'C1269765', SEMTYPES='[QLCO]', TRIGGER='["ASSI-STED (QUALIFIER VALUE)"-TX-1-"HELP"-VERB-0]']. This is a messed up world.

Metamap finds trigger words (i.e. depressed) and then maps it to a UMLS concept that has a preferred name (Depressed mood), a Concept Unique Identifier or CUI (C0344315), a semantic type (fndg or Finding) and some properties of the trigger itself.