Deep Learning for Suicide and Depression Identification with Unsupervised Label Correction

Ayaan Haque[†,*1], Viraaj Reddi[†, 1], Tyler Giallanza[‡]
[†] Saratoga High School, Saratoga, CA, USA
[‡] Department of Psychology & Princeton Neuroscience Institute, Princeton University, Princeton, NJ, USA

Abstract

Early detection of suicidal ideation in depressed individuals can allow for adequate medical attention and support, which in many cases is life-saving. Recent NLP research focuses on classifying, from a given piece of text, if an individual is suicidal or clinically healthy. However, there have been no major attempts to differentiate between depression and suicidal ideation, which is an important clinical challenge. Due to the scarce availability of EHR data, suicide notes, or other similar verified sources, web query data has emerged as a promising alternative. Online sources, such as Reddit, allow for anonymity that prompts honest disclosure of symptoms, making it a plausible source even in a clinical setting. However, these online datasets also result in lower performance, which can be attributed to the inherent noise in web-scraped labels, which necessitates a noise-removal process. Thus, we propose SDCNL, a suicide versus depression classification method through a deep learning approach. We utilize online content from Reddit to train our algorithm, and to verify and correct noisy labels, we propose a novel unsupervised label correction method which, unlike previous work, does not require prior noise distribution information. Our extensive experimentation with multiple deep word embedding models and classifiers display the strong performance of the method in a new, challenging classification application. We make our code and dataset available at https://github.com/ayaanzhaque/SDCNL

Keywords: Suicide/Depression, Noisy Labels, Deep Learning, Online Content, Natural Language Processing, Unsupervised Learning

1. Introduction

Depression remains among the most pressing issues worldwide, especially in developed and emerging countries. According to the World Health Organization (WHO), 322 million people worldwide suffer from depression [1]. Depression can often progress to suicidal ideation and attempted suicide if left unaddressed.

Given the severity of the issue, diagnosis of depression, and identification of when a depressed individual is at risk of attempting suicide, is an important problem at both the individual and population level. Many existing methods for detecting suicidal ideation rely on data from sources such as questionnaires, Electronic Health Records (EHRs), and suicide notes [2]. However, acquiring data in such formats is challenging and ultimately results in limited datasets, complicating attempts to accurately automate diagnosis.

Conversely, as the Internet and specifically social media have grown, online forums have developed into popular resources for struggling individuals to seek guidance and assistance. These forums are extensive and constantly expanding due to strong user support, and information on these forums is publicly accessible free of charge, suggesting the possibility of using web-scraped data from such forums to build automated systems for diagnosing mental health disorders.

Because these data present a promising alternative to traditional diagnostic resources, especially for machine learning techniques that require large datasets to be trained efficiently, a growing number of studies are using this data for diagnostic purposes. For example, [3] compiled and created a dataset of 3 million tweets from Twitter, and [4] used this dataset to identify depression with 86% accuracy (see [2] for a review of related methods).

In particular, Reddit has emerged as an important data source for diagnosing mental health disorders [5]. Reddit is an online social media platform similar in structure to forums,

[1] Authors contributed equally
[*] ayaanzhaque@gmail.com

in which users form communities with defined purposes referred to as subreddits. Certain subreddits are oriented towards discussing dealing with mental health, such as r/depression and r/SuicideWatch. Within these mental health subreddits, users share what has been troubling them and speak openly about their situation. While many social media platforms allow for anonymity, Reddit specifically allows users to create alternate and discardable accounts to ensure their privacy. This anonymity promotes disclosure and allows those with little support system in real life to receive support online [6]. The wide user base, honesty of these online settings, and moderated screening of these posts to ensure legitimacy provides an unprecedented opportunity for computationally analyzing mental health issues on a large scale, as demonstrated by [7].

These efforts support the validity of using such data sources to identify patients with mental health struggles such as anxiety and depression. However, there remains little work focused on detecting when individuals with underlying mental health struggles such as depression are at risk of attempting suicide. This represents an important clinical challenge, both for the advancement of how depression is treated and for implementing interventions [8, 9]. A machine learning solution specifically designed to distinguish between suicidality and other potentially less serious forms of depression is needed.

A possible reason for the lack of a current solution is that distinguishing between suicidality and depression is a more fine-grained task than distinguishing between, for example, suicidal and healthy behavior. Online data has traditionally been difficult to use in such fine-grained situations, because labels for such data are often unreliable given their informal nature and lack of verification. In particular, labeling data based on subreddit relies on self-reporting, since each user chooses which subreddit they feel best reflects their mental state; thus, they may over or under report their diagnosis. This concept is referred to as *noisy labels* as there is a potential for certain labels to be corrupted. Estimates show that noisy labels can degrade anywhere from 10% to 40% of the dataset [10, 11], presenting serious challenges for machine learning algorithms.

Current attempts to address the noisy label problem can be categorized into three notable groups: noise-robust methods, noise-tolerant methods, and data cleaning [10, 12]. Noise-robust and noise-tolerant methods both involve modifications to the architecture and/or loss function of the machine learning algorithm used. Noise-robust approaches rely on algorithms that are naturally less sensitive to noise (e.g. lower dimensional or regularized algorithms), whereas noise-tolerant methods directly model the noise during training. Although both approaches have received considerable attention in the image-processing domain [10, 13], these methods do not transfer to NLP algorithms. In the NLP domain, there have been a few recently proposed noisy label methods [14–17]. However, the proposed methods have a number of limitations that make their application unsuitable for the present task. For example, some methods utilize a smaller set of trusted data to correct a larger set of noisy data [16, 17], which is infeasible for our application due to the lack of trusted data or knowledge of which posts can be trusted. Other methods require training a network directly and end-to-end from corrupted labels [14, 15], which both requires a relatively large amount of data and is less capable of leveraging transfer learning from pre-trained, state-of-the-art models [18–20].

Data cleaning methods are more suitable for the present task. However, most existing label cleaning methods make assumptions about or require knowledge on the distribution of noise in the dataset [21–23]. In our use-case, where there is no prior knowledge of the noise distribution, an unsupervised method, such as clustering, is required. Although there are a few methods that use unsupervised clustering algorithms for noisy label learning [24, 25], none of these correct labels. Rather, they train a model to be robust to noise through instance weighting or exclusion. These methods would be problematic for our task; due to the high noise proportion, weighting or removing a high volume of data would be

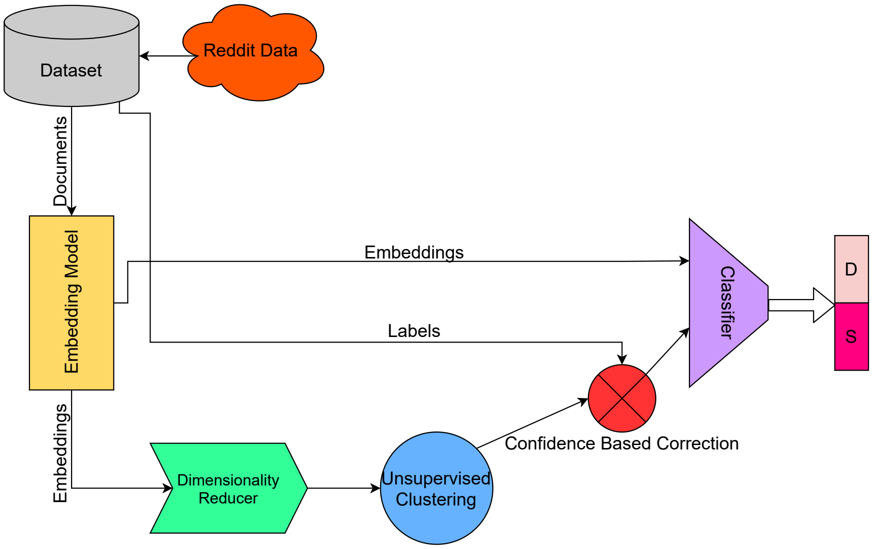

Figure 1. Schematic of the SDCNL pipeline used for classification of suicide vs depression and noisy label correction via unsupervised learning.

damaging to performance, especially for deep learning methods that require large amounts of data. Thus, the present task requires an unsupervised method for data cleaning that utilizes label correction rather than elimination. To the best of our knowledge, there are no current methods which perform label correction using unsupervised clustering methods, and particularly not in the NLP domain.

In this paper, we present SDCNL to address the unexplored issue of classifying between depression and more severe suicidal tendencies using web-scraped data. In particular, we leverage Reddit data, develop a novel label correction method to remove inherent noise in the data using unsupervised learning, and develop a deep-learning classifier based on pre-trained transformers [18–20]. Our primary contributions can be summarized as follows:

- An application of deep learning-based sentiment analysis for depression versus suicidal ideation classification, an important but unexplored clinical and computational challenge
- A novel, unsupervised label correction process for text-based data and labels which does not require prior noise distribution information, allowing for the use of mass online content
- Extensive experimentation and ablation on multiple datasets, demonstrating the success of our label correction and deep learning classification approach on the challenging proposed task

2. Methods

The SDCNL method is outlined in Figure 1. We begin by processing text data scraped from Reddit with word embedding models, which convert raw text into numerical representations called embeddings [26]. These embeddings are then processed with an unsupervised dimensionality reduction algorithm. This is a necessary procedure due to the nature of clustering algorithms, which do not perform well in high-dimensional domains [27]. The reduced embeddings are then inputted into a clustering-based algorithm which separates the embeddings into a pre-determined number of classes. This clustering algorithm is unsupervised, allowing it to be independent of noise in the labels. The class predictions of the algorithm are treated as an alternate set of labels, and these predicted labels are compared against the ground-truth labels using a confidence-based thresholding procedure in order to correct the ground-truth labels. The corrected set of labels are then used to train a deep neural classifier in a supervised fashion.

2.1. **Embedding Models**

Our framework initially utilizes word embedding models to convert raw documents, which in our case are referred to as posts, to numerical word embeddings. Our proposed method can be used with any embedding models, but given our task, we require greater-than-word text embedding models optimized to work with phrases, sentences, and paragraphs. We experiment with 3 state-of-the-art transformers: Bidirectional Encoder Representations from Transformers (BERT) [18], Sentence-BERT [19], and the Google Universal Sentence Encoder (GUSE) [20]. BERT is a state-of-the-art, bidirectionally trained transformer that achieves high performance on various benchmark NLP tasks, and outputs a 768 × 512 dimensional vector of embeddings. Sentence-BERT is an extension of the original BERT architecture that is retrained and optimized for longer inputs and better performance during clustering, and it outputs a 768 × 1 dimensional vector. GUSE is a transformer also trained and optimized for greater-than-word length text, but rather returns a 512 × 1 dimensional vector.

Some classifiers require word level representations for embeddings, while others require document level representations. BERT outputs both multi-dimensional word level embeddings as well as document level embeddings, which are provided by CLS tokens. Depending on what the classifier requires, we vary the inputted embeddings to match the classifier's requirement. In addition, we also experiment on three vectorizers as baselines: Term Frequency–Inverse Document Frequency (TFIDF), Count Vectorizer (CVec), and Hashing Vectorizer (HVec). These vectorizers are standard NLP embedding models for text vectorization and information retrieval.

2.2. **Label Correction**

To address the issue of label noise for our task, we propose an unsupervised label correction method. We initially feed our word embeddings through a dimensionality reduction algorithm to convert the high-dimensional features outputted by the embedding models to lower-dimensional representation. Due to the nature of most clustering algorithms, high-dimensional data typically results in subpar performance and poorly separated clusters, a phenomenon known as the "Curse of Dimensionality" [27]. Thus, representing the data in lower dimensions is a necessary procedure. We experiment with three separate dimensionality reduction algorithms: Principal Component Analysis (PCA), Deep Neural Autoencoders [28], and Uniform Manifold Approximation and Projection (UMAP) [29]. PCA is a common reduction algorithm that extracts the most important information from a matrix of numerical data and represents it as a set of new orthogonal variables. Autoencoders, which have recently been gaining attention, use an unsupervised neural network to compress data into a low-dimensional space, and then reconstruct it while retaining the most possible information. The output of the encoder portion is used as the reduced embeddings. UMAP produces a graph from high-dimensional data and is optimized to generate a low-dimensional graph as similar to the input as possible. UMAP is specifically effective for high-dimensional data, as it has improved preservation of global structure and increased speed.

After reducing the dimensions of our word embeddings, we use clustering algorithms to separate them into two distinct clusters, which allow us to assign new labels to each post. We leverage clustering algorithms because of their unsupervised nature; this is critical because we have no prior knowledge regarding the noise distribution in the labels, requiring a clustering procedure which is independent of the web-scraped labels. We use the Gaussian

Mixture Model (GMM) as our clustering algorithm. A GMM is a parametric probability density function used as a model of the probability distribution of continuous measurements in order to cluster given data using probabilities. As a baseline, we use K-Means clustering; K-means attempts to divide n observations into k clusters, such that each observation is assigned to the cluster with the closest mean, and the clusters minimize within-cluster distance while maximizing between-cluster distance. Lastly, we additionally attempt to circumvent the dimensionality reduction requirement due to the reduced information from the embeddings. We use subspace clustering via spectral clustering [30, 31], which specifically allows unsupervised clustering of high-dimensional data by identifying clusters in different sub-spaces within a dataset.

Each word embedding is now associated with two labels: the original labels based on the subreddit, which are the ground-truth labels, and the new labels resulting from unsupervised clustering. We subsequently leverage a confidence-based thresholding method to correct the ground-truth labels. If the clustering algorithm predicts a label with a probability above τ, a tuned threshold, the ground-truth label is replaced with the predicted label; otherwise, we assume the ground-truth label. The tuning threshold ensures only predicted labels with high confidence are used to correct the ground-truth, preventing false corrections. Finally, the correct set of labels are paired with their respective post. We obtain class probabilities using the GMM algorithm, with K-means as a baseline. Note that our label correction method can be used in any NLP domain or even in other fields, such as the imaging field.

2.3. **Classification**

After determining the final set of corrected labels, we train our deep learning networks to determine whether the posts display depressive or suicidal sentiment. This ultimately is the final application of our paper. Similar to the embedding process, any classifier can be used in place of the ones we tested. However, we aim to prove that deep neural classifiers are effective for our proposed task.

For experimentation, we tested four deep learning algorithms: a dense neural network, a Convolutional Neural Network (CNN), a Bidirectional Long Short-Term Memory Neural Network (BiLSTM), a Gated Recurrent Unit Neural Network (GRU). For baselines, we evaluated three standard machine learning models: Logistic Regression (LogReg), Multinomial Naive Bayes (MNB), and a support-vector machine (SVM).

2.4. **Datasets**

2.4.1. **Web-Scraped Depressed vs Suicidal Dataset**

We develop a primary dataset based on our task of suicide or depression classification. This dataset is web-scraped from Reddit. We collect our data from subreddits using the Python Reddit API [2]. We specifically scrape from two subreddits, r/SuicideWatch[3] and r/Depression[4]. The dataset contains 1,895 total posts. We utilize two fields from the scraped data: the original text of the post as our inputs, and the subreddit it belongs to as labels. Posts from r/SuicideWatch are labeled as suicidal, and posts from r/Depression are labeled as depressed. We make this dataset and the web-scraping script available in our code.

2.4.2. **Reddit C-SSRS Dataset**

Furthermore, we use the Reddit Suicide C-SSRS dataset [32] to verify our label correction methodology. The C-SSRS dataset contains 500 Reddit posts from the subreddit r/depression.

[2] https://www.reddit.com/wiki/api

[3] https://www.reddit.com/r/SuicideWatch/

[4] https://www.reddit.com/r/depression/

These posts are labeled by psychologists on a five point scale according to guidelines established in the Columbia Suicide Severity Rating Scale, which progress according to severity of depression. As this dataset is clinically verified and labeled, it is an adequate dataset to validate the label correction method, especially since it is from the same domain of mental health.

2.4.3. Additional Datasets

To further validate the label correction method, we use the IMDB large movie dataset, a commonly used NLP benchmark dataset [33]. The dataset is a binary classification task which contains 50,000 highly polar movie reviews. We use a random subset of samples for validation.

For comparison of our method against other related tasks and methods, we build a dataset for binary classification of clinically healthy text vs suicidal text. We utilize the two subreddits r/CasualConversation and r/SuicideWatch. r/CasualConversation is a subreddit of general conversation, and has generally been used by other methods as data for a clinically healthy class [34].

3. Experimental Results

3.1. Implementation Details

For all datasets, we set aside 20% of the dataset as an external validation set. The deep learning models were implemented with Tensorflow, and the rest of the models were implemented with Sci-Kit Learn. We trained the deep learning models with the Adam optimizer and used a binary cross-entropy loss function. Based on tuning experiments (not included in the paper), we set τ to 0.90 for all experiments, but similar values yielded similar performance. For classification accuracy, we use five metrics: Accuracy (Acc), Precision (Prec), Recall (Rec), F1-Score (F1), and Area Under Curve Score (AUC). Model-specific hyperparameters are included in the code.

3.2. Label Correction Performance

To evaluate our clustering performance, we present both the accuracy of the clustering algorithm at correcting noisy labels as well as classification performance after label correction. Classification on an clean test set is expected to decrease as training labels become noisier [12]. Therefore, we contend that if after label correction, the classification accuracy of our algorithm increases, the correction method is effective.

Importantly, because our proposed task uses a web-scraped dataset, the labels are not clinically verified. This unfortunately means evaluating the correction rate of noisy labels is impossible because we do not have the true label. Therefore, we perform evaluation of clustering accuracy on two datasets. We first evaluate on the benchmark IMDB dataset to demonstrate the value of our method in a general setting. We then evaluate on the C-SSRS dataset, which contains clinically verified labels, to demonstrate the effectiveness of the proposed label correction method in our specific domain.

3.2.1. Clustering Performance

To evaluate the performance of clustering, we inject noise into the label set at different rates. We corrupt 10-40% of the dataset at both uniform and imbalanced rates, as the noise rate of labels in real-world datasets are estimated to be 8% to 38.5%. These noise levels are also standard for other noisy label papers [10]. We then evaluated the performance of the clustering algorithms at correcting the noisy labels.

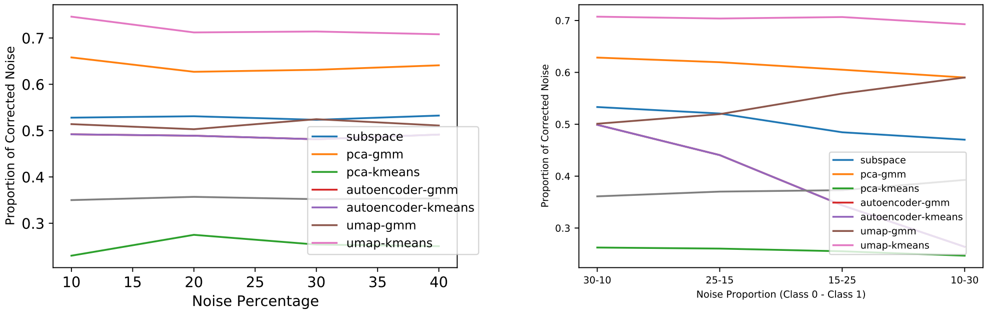

Figure 2. Correction rates of the label correction algorithms at different noise rates on the **IMDB** dataset. The left figure displays correction rates with uniform injections of noise, and the right figure displays correction rates with class-weighted injections of noise. For example, we split by 30%-10% or 25%-15%.

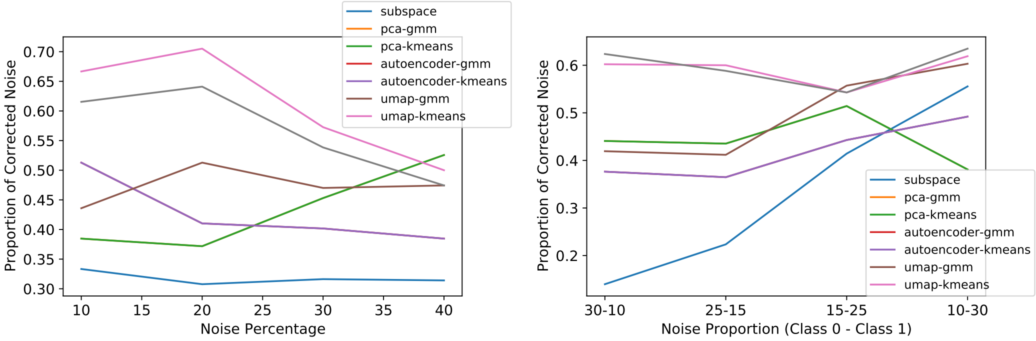

Figure 3. Correction rates of the label correction algorithms at different noise rates on **C-SSRS**. The left figure displays correction rates with uniform injections of noise, and the right figure displays correction rates with class-weighted injections of noise. For example, we split by 30%-10% or 25%-15%.

As seen in Figure 2 and 3, our noise correction method is able to consistently remove >50% of injected noise while remaining below a <10% false-correction rate on both datasets, and the performance does not degrade heavily at higher noise percentages, which is challenging to achieve [12].

The SDCNL label correction is successful on both the IMDB dataset, which shows the generalizability of the method, as well as the C-SSRS dataset, proving its ability in our specific domain and task. The best combinations of reduction and clustering algorithms are umap-kmeans and umap-gmm, so for our final label correction tests, we used those two combinations, with K-Means as the baseline and GMM as the proposed method. To our best knowledge, almost all noisy label correction methods do not evaluate correction rates but rather evaluate performance on classification accuracy after correction, as it allows for comparison to other noisy label methods that do not use label correction. However, because most recent noisy label methods are in the image domain, drawing comparisons to related work is unfeasible.

3.2.2. Classification Performance after Label Correction

Lastly, to demonstrate the effectiveness of the label correction method, we train a classifier on noisy C-SSRS data and validate on a separate C-SSRS test set which has no noise. We

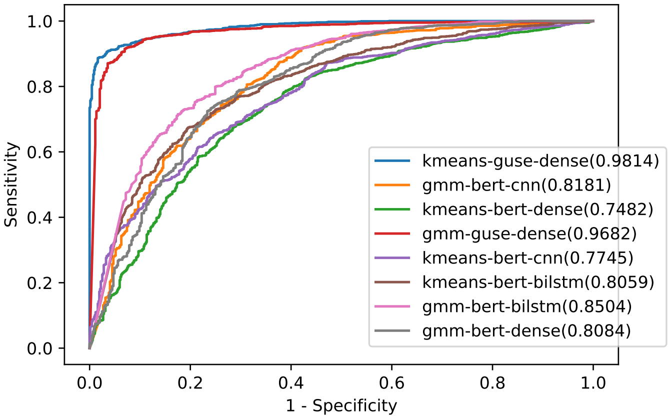

Figure 4. ROC curves of model performance after using label correction. The 4 best combination of models with the two final label correction methods are shown (GMM vs K-Means). UMAP is used to reduce the dimensions of the embeddings.

then use our label correction method to correct the same set of noisy labels, train the model with the correction labels, and validate on the same unmodified test set.

Model	Accuracies per Task (%)	
	Noisy	**Corrected**
guse-dense	57.97	70.63
bert-dense	48.86	70.13
bert-bilstm	56.71	68.35
bert-cnn	55.70	70.13

Table 1. Accuracy comparison after injecting randomized noise into C-SSRS labels to train a classifier (left) against using the label correction method to remove the artificial noise and training the classifier after (right).

We show that accuracy improves markedly after using our label correction method, as there is at least a 11% increase (Table 1). As demonstrated, because our label correction process works on a dataset in the same domain, it is an effective method for cleaning noisy labels in NLP and for our task.

Moreover, as seen in Figure 4, using a probability threshold impacts performance, proving that using a threshold is an important factor in ensuring the corrected labels are accurate. Thus, we finalize on the thresholding method for our final model.

3.3. Classification Performance

For classification performance, we present three primary evaluations. We begin with an extensive experimentation process to evaluate numerous combinations of word embedding models and classifiers in order to demonstrate the greater performance of deep learning-based models. We then present the difficulty of our proposed task against the traditionally researched task of suicidal or clinically healthy text classification. We conclude this section with the final performance of our algorithm with the four best model combinations after label correction to determine the final proposed model.

3.3.1. **Deep Learning Performance**

Metrics (%)	Model Combinations			
	guse-dense	bert-dense	bert-bilstm	bert-cnn
Acc	**72.24**	70.50	71.50	72.14
Rec	**76.37**	71.92	67.77	73.99
Prec	71.38	70.77	**74.28**	72.18
F1	**73.61**	71.25	70.70	72.92
AUC	**77.76**	75.43	77.11	76.35

Table 2. Performance of four best initial combinations of embedding models and classifiers are shown.

After performing all experiments, we determined the four strongest combinations to perform the remainder of the tests. These combinations are trained on the primary suicide vs depression dataset with uncorrected labels. The complete results can be viewed in the Appendix A in the supplemental. The performance of the four strongest models are shown in Table 2. The combinations are: BERT embeddings with a CNN (bert-cnn), BERT with a fully-dense neural network (bert-dense), BERT with a Bi-LSTM neural network (bert-bilstm), and GUSE with a fully-dense neural network (guse-dense). This proves the importance of our contribution, as all of the transformers and classifiers are deep learning models, substantially outperforming the baselines. For all future experiments, we use the four models above.

3.3.2. **Comparison to Other Tasks**

While there is extensive research on NLP text-based approaches to suicide detection, there is none for our specific task of low-risk depression versus suicidal ideation. We performed an additional test of our proposed model by testing the standard task of classifying suicide versus clinically healthy.

Figure 5 and Table 3 display that on identical models, the commonly-researched task achieved substantially stronger baseline performance compared to our task; therefore, our baseline evaluation metrics should correspondingly be lower. This difficulty of task also proves the motivation; for both psychologists and computational approaches, our task is incredibly challenging, making it valuable in a true clinical setting.

3.3.3. **Final Evaluation**

We evaluated the classification performance of SDCNL on the collected Reddit dataset. We generated ground-truth labels for this dataset using the proposed label correction method.

Metrics (%)	Proposed				**Standard**			
	guse-dense	bert-dense	bert-bilstm	bert-cnn	guse-dense	bert-dense	bert-bilstm	bert-cnn
Acc	72.24	70.50	71.50	72.14	92.28	57.46	92.78	93.11
Prec	76.37	71.92	67.77	73.99	93.56	80.70	92.16	92.26
Rec	71.38	70.77	74.28	72.18	92.46	58.28	94.54	95.07
F1	73.61	71.25	70.70	72.92	93.00	67.39	93.35	93.63
AUC	77.76	75.43	77.11	76.35	96.65	54.91	97.24	96.49

Table 3. Comparison of classification metrics between conventionally researched task of suicide vs clinically healthy against our proposed task of suicide vs depression. The better performing category is bolded. The standard task performs far better on the same models, highlighting how our proposed task is more difficult to categorize.

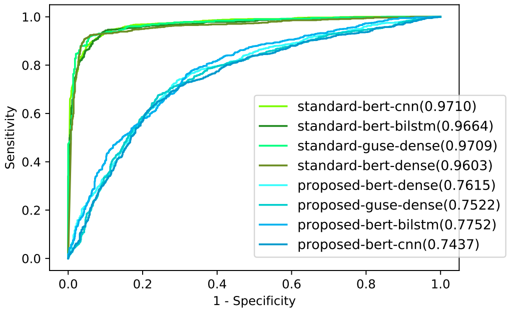

Figure 5. ROC curves of model performance from four best models on our task (proposed) against the conventional suicide vs healthy task (standard).

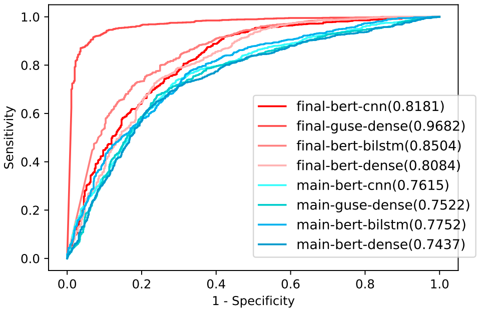

Figure 6. ROC curves of performance of top 4 models with label correction (red) against the same models without label correction (blue).

Metrics (%)	UMAP-KMeans				UMAP-GMM			
	guse-dense	bert-dense	bert-bilstm	bert-cnn	***guse-dense***	bert-dense	bert-bilstm	bert-cnn
Acc	**92.61**	73.56	75.15	74.72	**93.08**	83.74	84.16	84.59
Prec	**93.61**	83.18	84.02	87.38	94.76	**95.51**	93.10	95.38
Rec	**94.85**	77.66	79.00	76.65	**96.16**	85.08	87.09	86.05
F1	**94.22**	80.25	81.19	81.64	**95.44**	89.99	89.99	90.45
AUC	**98.18**	76.83	80.93	78.69	**96.88**	81.97	85.08	82.91

Table 4. Final evaluation of the classification performance after using the two label correction methods, with and without a thresholding scheme. The best performances for each noise removal method are bolded. Best overall model is bolded and italicized.

To provide a complete test of our model, it would be more valid to use labels provided by a mental health professional; however, no such dataset exists for our task.

We contend that, since we demonstrated that the proposed label correction method is effective on the C-SSRS and IMDB datasets, the use of the label correction method on the Reddit dataset is valid.

Table 4 displays the final performance of the models with both threshold label correction (GMM) and without (K-Means). We validate the importance of the thresholding component, as with thresholding, all metrics are significantly improved over the K-Means baseline. Moreover, as displayed in Figure 6, our label correction method improves the ROC curve and yields a much higher AUC value. This proves the importance and effectiveness of our label correction process, as simply by correcting the labels in the test set where noise is expected to be high, we are able to achieve substantially higher performance. Our final proposed combination uses GUSE as the embedding model, a fully-dense network for the classifier, and corrects labels using a UMAP for dimensionality reduction and a GMM for the clustering algorithm.

4. Conclusion

In this paper, we present SDCNL, a novel method for classification of depressive sentiment vs suicidal ideation by leveraging deep learning and unsupervised noisy label correction. Our novel method of label correction using unsupervised clustering effectively removes high-volumes of noise from both benchmark and domain-specific datasets, allowing for the use of large-scale, web-scraped datasets. Our extensive experimentation and ablative results highlight the effectiveness of our proposed model and the potential of our model for real diagnostic application.

The applied setting of our system is to provide professionals with a supplementary tool for individual patient diagnosis, as opposed to solely being a screening method on social media platforms. SDCNL could be used by professional therapists as a "second opinion", friends and family as a preliminary screening for loved ones, or even on social media platforms to identify at-risk users. We contend that our method is applicable in a clinical setting given Reddit's propensity for honest user disclosure, especially considering the ability of our label correction method to remove substantial amounts of noise from the labels. As opposed to models trained on EHRs or suicide notes, which often include limited amounts of data, leveraging online content allows for the use of data-hungry deep learning models, which we prove can be quite effective.

References

[1] W. H. Organization. *Depression and other common mental disorders: global health estimates*. Technical documents. 2017, 24 p.

[2] S. Ji et al. "Suicidal ideation detection: A review of machine learning methods and applications". In: *IEEE Transactions on Computational Social Systems* (2020).

[3] G. Coppersmith et al. "CLPsych 2015 shared task: Depression and PTSD on Twitter". In: *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. 2015, pp. 31–39.

[4] M. Nadeem. "Identifying depression on Twitter". In: *arXiv preprint arXiv:1607.07384* (2016).

[5] A. N. Medvedev, R. Lambiotte, and J.-C. Delvenne. "The anatomy of Reddit: An overview of academic research". In: *Dynamics on and of Complex Networks*. Springer. 2017, pp. 183–204.

[6] M. De Choudhury and S. De. "Mental health discourse on reddit: Self-disclosure, social support, and anonymity". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 8. 2014.

[7] J. Shen and F. Rudzicz. "Detecting Anxiety through Reddit". In: Jan. 2017, pp. 58–65. DOI: 10.18653/v1/W17-3107.

[8] J. Bering. *Suicidal: Why we kill ourselves*. University of Chicago Press, 2018.

[9] C. Leonard. "Depression and suicidality." In: *Journal of Consulting and Clinical Psychology* 42.1 (1974), p. 98.

[10] H. Song et al. "Learning from noisy labels with deep neural networks: A survey". In: *arXiv preprint arXiv:2007.08199* (2020).

[11] H. Song, M. Kim, and J.-G. Lee. "SELFIE: Refurbishing Unclean Samples for Robust Deep Learning". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 5907–5915.

[12] B. Frénay and M. Verleysen. "Classification in the Presence of Label Noise: A Survey". In: *Neural Networks and Learning Systems, IEEE Transactions on* 25 (May 2014), pp. 845–869. DOI: 10.1109/TNNLS.2013.2292894.

[13] G. Algan and I. Ulusoy. "Image classification with deep learning in the presence of noisy labels: A survey". In: *Knowledge-Based Systems* (2021), p. 106771.

[14] I. Jindal et al. *An Effective Label Noise Model for DNN Text Classification*. 2019. arXiv: 1903.07507 [cs.LG].

[15] S. Zheng et al. "Error-bounded correction of noisy labels". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11447–11457.

[16] D. Hendrycks et al. "Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018, pp. 10456–10465. URL: https://proceedings.neurips.cc/paper/2018/file/ad554d8c3b06d6b97ee76a2448bd7913-Paper.pdf.

[17] G. Zheng, A. H. Awadallah, and S. Dumais. "Meta label correction for learning with weak supervision". In: *arXiv preprint arXiv:1911.03809* (2019).

[18] J. Devlin et al. "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, June 2019, pp. 4171–4186.

[19] N. Reimers and I. Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL].

[20] D. Cer et al. "Universal Sentence Encoder for English". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2018, pp. 169–174.

[21] Z. Jiang et al. "Learning from Noisy Labels with Noise Modeling Network". In: *arXiv preprint arXiv:2005.00596* (2020).

[22] D. Hendrycks et al. "Using trusted data to train deep networks on labels corrupted by severe noise". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018, pp. 10477–10486.

[23] A. J. Bekker and J. Goldberger. "Training deep neural-networks based on unreliable labels". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 2682–2686. DOI: 10.1109/ICASSP.2016.7472164.

[24] C. Bouveyron and S. Girard. "Robust supervised classification with mixture models: Learning from data with uncertain labels". In: *Pattern Recognition* 42.11 (Nov. 2009), pp. 2649–2658.

[25] U. Rebbapragada and C. E. Brodley. "Class noise mitigation through instance weighting". In: *European Conference on Machine Learning*. Springer. 2007, pp. 708–715.

[26] Y. Li and T. Yang. "Word embedding for understanding natural language: a survey". In: *Guide to big data applications*. Springer, 2018, pp. 83–104.

[27] M. Steinbach, L. Ertöz, and V. Kumar. "The challenges of clustering high dimensional data". In: *New directions in statistical physics*. Springer, 2004, pp. 273–309.

[28] W. Wang et al. "Generalized autoencoder: A neural network framework for dimensionality reduction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014, pp. 490–497.

[29] L. McInnes, J. Healy, and J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[30] U. Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[31] L. Parsons, E. Haque, and H. Liu. "Subspace Clustering for High Dimensional Data: A Review". In: *SIGKDD Explor. Newsl.* 6.1 (June 2004), pp. 90–105. ISSN: 1931-0145. DOI: 10.1145/1007730.1007731. URL: https://doi.org/10.1145/1007730.1007731.

[32] *Reddit C-SSRS Suicide Dataset*. Zenodo, May 2019. DOI: 10.5281/zenodo.2667859. URL: https://doi.org/10.5281/zenodo.2667859.

[33] A. L. Maas et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

[34] N. Schrading et al. "An analysis of domestic abuse discourse on reddit". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2577–2583.

Appendix A. Baseline Model Evaluations

Appendix A displays the full experimentation of baseline results for 6 transformers and 7 classification algorithms (Table 5). The results are evaluated with 5 metrics: Accuracy (Acc), Precision (Prec), Recall (Rec), F1 score (F1), and Area Under Curve (AUC). With these results, we selected the four strongest performing combinations of models to perform the remainder of the experimentation.

Appendix B. Vectorizers and Clustering

Appendix B shows how the number of extracted features from the vectorizers was chosen (Figure 7). The AUC scores from the MNB classifier after using the vectorizers with different

Table 5. Performance of all 42 combinations. 7 classifiers and 6 embedding models are used, with 4 deep classifiers and 3 deep embedding models.

Embedding Model		Classifiers						
	Metrics	CNN	Dense	BiLSTM	GRU	MNB	SVM	LogReg
BERT	Acc	72.14	70.50	71.50	71.50	57.78	68.07	68.60
	Rec	73.99	71.92	67.78	68.91	53.37	73.58	70.98
	Prec	72.18	70.77	74.28	73.86	59.54	66.98	68.50
	F1	72.92	71.25	70.70	71.05	56.28	70.12	69.72
	AUC	76.35	75.43	77.11	76.66	54.21	55.43	54.72
SentenceBERT	Acc	68.65	68.87	69.55	70.77	59.37	68.34	63.85
	Rec	73.37	74.61	67.98	67.36	46.63	73.06	69.95
	Prec	67.88	67.94	71.22	73.35	63.83	67.46	63.08
	F1	70.40	70.82	69.41	70.01	53.89	70.15	66.34
	AUC	73.52	73.70	74.00	74.99	56.13	53.12	51.33
GUSE	Acc	72.66	72.24	72.82	73.19	69.39	71.50	71.50
	Rec	78.96	76.37	78.03	77.10	67.36	75.65	74.61
	Prec	70.79	71.38	71.36	72.31	71.04	70.53	70.94
	F1	74.62	73.61	74.49	74.52	69.15	73.00	72.73
	AUC	77.82	77.76	77.41	76.33	47.68	49.47	50.75
TFIDF	Acc	67.12	69.28	67.97	67.92	69.39	70.45	69.13
	Rec	81.35	67.78	72.33	74.61	76.68	74.09	73.58
	Prec	65.98	70.73	67.41	66.61	67.58	69.76	68.27
	F1	71.89	69.19	69.64	70.21	71.85	71.86	70.83
	AUC	70.04	75.23	71.70	72.23	51.65	51.38	51.35
CountVec	Acc	69.18	68.92	68.13	67.86	66.75	65.43	63.32
	Rec	82.07	73.47	75.23	74.82	72.54	69.43	66.84
	Prec	65.91	68.32	66.62	66.41	65.73	65.05	63.24
	F1	73.06	70.61	70.59	70.31	68.97	67.17	64.99
	AUC	74.44	73.34	72.66	72.30	51.47	51.08	47.35
HashVec	Acc	67.86	67.44	65.49	65.17	65.96	66.23	66.75
	Rec	71.50	69.02	66.74	68.19	69.43	69.95	72.54
	Prec	67.58	67.79	66.34	65.30	65.69	65.85	65.73
	F1	69.27	68.31	66.17	66.47	67.51	67.84	68.96
	AUC	71.47	71.63	68.98	69.58	52.94	54.06	52.85

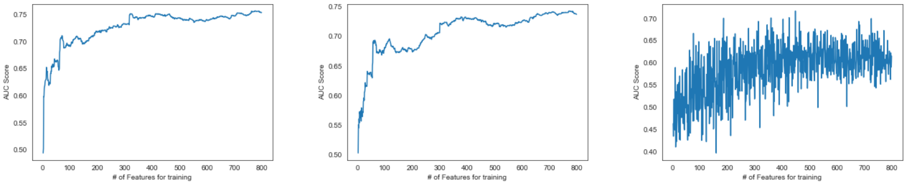

Figure 7. Area Under Curve (AUC) values at different number of extracted word embeddings/features from the three vectorizers (TFIDF, CVec, HVec) inputted into Bayesian classifier. AUC plateaus at around 400 features, so we used 768 features to be consistent with other word embedding models.

number of embeddings are shown. After around 400 features for each model, the performance converges, so we finalized on extracting 768 features for consistency.

Figure 8 visualizes the clustering of a GMM using BERT embeddings with PCA reduction.

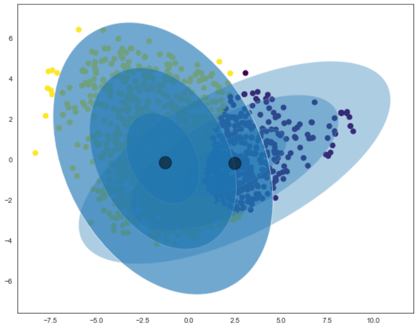

Figure 8. GMM clustering using BERT embeddings and PCA reduction to 2 dimensions show the difficulty of the clustering task, as there is little variety in the clusters and they heavily overlap. We use co-variance type "full".