How Well Can BERT Detect Suicidal Ideation?

Hugh Cheng | hughcheng1@gmail.com
Harvard-Westlake, Class of 2025
USC Viterbi Department of Computer Science, SHINE 2023

USC Viterbi School of Engineering

SHINE Summer High School Intensive in Next-Generation Engineering

Introduction

INK Lab at USC

Research in the Intelligence and Knowledge Discovery **(INK)** Lab spans across multiple fields in artificial intelligence. My lab mentor, **Hirona Arai,** is doing research in natural language processing (NLP). The lab is run by my professor, **Xiang Ren.**

Professor Ren's current research focuses on giving computers the common sense humans have learned from years of experience interacting with the real world.

Purpose and Setup

Research Goal and Impact

Suicide is the 11th leading cause of death in the US. In the last six months, my high school lost three students to suicide, each death leaving a devastating blow to the community. According to the CDC, suicide rates have increased by 36% from 2000 to 2021 *(Figure 1)*.[1]

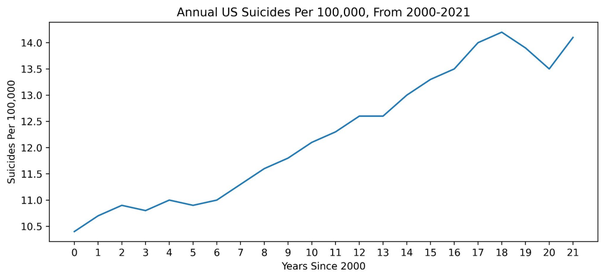

Figure 1. *Suicide rates are rising*

In this project, I trained and tuned Google's **BERT** (Bidirectional Encoder Representations from Transformers) to detect signs of people considering suicide in text. Hopefully, this tool can be used to provide aid to victims of depression.

Dataset

The original dataset[2] used is a collection of 230,000 Reddit posts, half of which are taken from **r/suicidewatch** (a subreddit for people considering suicide to share their thoughts and emotions) and the other half taken from **r/teenagers** (to represent non-suicidal text). Due to Google Colab's built-in RAM and GPU limits, I only trained the model on 6,000 posts.

Baseline

Tuning Naive Bayes

For my baseline, I used a **Naive Bayes Classifier.** This model reads each token (word) separately, assuming that they are independent of each other, and calculates a classification based on the frequency of each token in the text. I tuned this model's smoothing hyperparameter alpha, as seen in *Figure 2:*

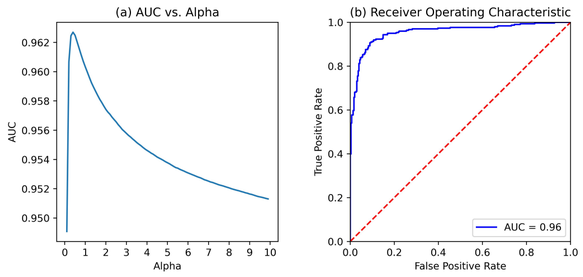

Figure 2. *(a) Tuning alpha on AUC (b) Optimized ROC Curve*

The tuned baseline achieves an AUC of 0.96, with an accuracy of **90.0%.**

BERT ALERT!!

Why BERT?

BERT[3] has been pre-trained on a large collection of English text, allowing it to grasp grammar, sentence structure, and context that is inaccessible to the Naive Bayes model. As a **masked language model** (MLM), BERT randomly replaces tokens with the [MASK] token during training, and predicts what the missing word was (*Figure 3).* The [CLS] token at the beginning of each sentence represents BERT's predicted classification for that sentence.

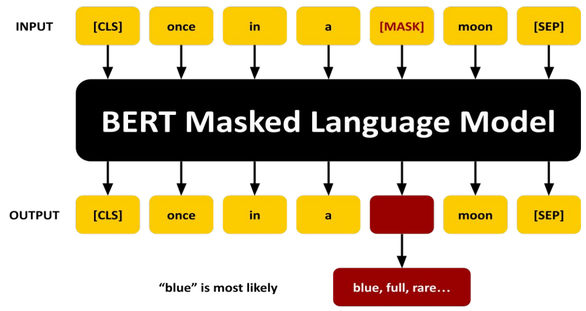

Figure 3. *BERT predicts masked words*

Tuning BERT

Initial Performance

The model was initially trained on 6,000 Reddit posts, with a default batch size of 32 and learning rate of 5e-5. It reached an AUC of 0.98 and validation accuracy of **94.8%** after 4 epochs, almost 5 points higher than the baseline performance.

Increasing Dataset Size

Are all 230,000 rows of the original dataset really necessary? In *Figure 4,* validation accuracy began to taper off after only around 40% of the 6,000 rows used in training. Using more data past this point would be expensive to run with diminishing returns.

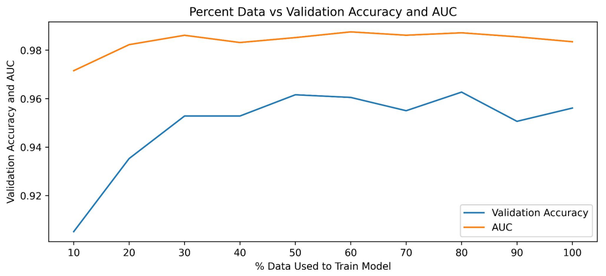

Figure 4. *Model performance plateaus*

Grid Search

To tune BERT, I used a **grid search** to exhaustively check many combinations of hyperparameters (batch size and learning rate). The maximal validation accuracy of **96.7%** was achieved at batch size 64 and learning rate 3e-5, after 8 epochs, as illustrated in *Figure 5.*

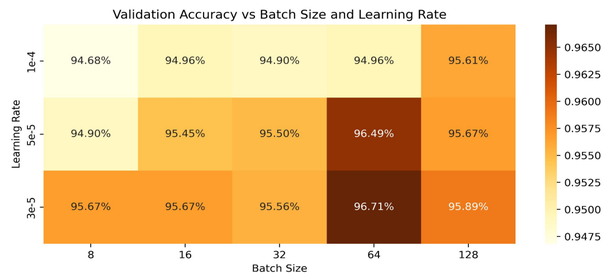

Figure 5. *Grid search results*

Using these hyperparameters, I similarly tuned the smoothing hyperparameter, epsilon. Changing epsilon had a marginal change on the model's performance, however, and the default epsilon of 1e-8 appeared to be optimal as well.

Conclusion

Results

The pre-trained BERT model is a significant improvement over the baseline, and after tuning it, I increased its validation accuracy by an additional 2 percent. With a threshold of 0.5, the model achieves a **95.4%** accuracy, **96.8%** recall, and **0.99** AUC on the remainder of the original dataset.

Next Steps

More powerful versions of BERT exist! I used **BERT-base,** which has 12 encoded layers, but another model called **BERT-large** has 24. A larger model might plateau at a higher accuracy, and could take advantage of the surplus of unused data.

Other tuning methods such as **RayTune** might be a better way to tune BERT. The basic grid search only covers a small space of hyperparameters.

Finally, the dataset should be expanded. The posts I used only constitute a small subset of suicidal and non-suicidal text that exists. Posts on r/suicidewatch only come from people who are specifically seeking responses from others, but the true goal of this model is to detect more subtle signs of suicide from any text. Such a dataset would need to be curated by a human being who is proficient in diagnosing mental health disorders.

Acknowledgements

I would like to thank Professor Ren and the SHINE Team for this incredible educational opportunity; my lab mentor Hirona for being an amazing and patient teacher; my lab partner Anushka for being a great friend; my center mentor Minsun for the games of skribbl.io; and my family for their support.

References

[1] "Suicide Data and Statistics," *Centers for Disease Control and Prevention,* Last modified (2023, May), Accessed on July 15, 2023, https://www.cdc.gov/suicide/suicide-data-statistics.html.
[2] Komati, Nikhileswar. (2021, May). Suicide Watch, Version 14. Retrieved July 10, 2023 from https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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),* pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.