# Cheap Ways of Extracting Clinical Markers from Texts

**Anastasia Sandu**
anastasiasandu777@gmail.com

**Teodor Mihailescu**
teomihailescu@yahoo.com

**Sergiu Nisioi**
sergiu.nisioi@unibuc.ro

Human Language Technologies Research Center
Faculty of Mathematics and Computer Science
University of Bucharest

## Abstract

This paper describes the work of the UniBuc Archaeology team for CLPsych's 2024 Shared Task, which involved finding evidence within the text supporting the assigned suicide risk level. Two types of evidence were required: highlights (extracting relevant spans within the text) and summaries (aggregating evidence into a synthesis). Our work focuses on evaluating Large Language Models (LLM) as opposed to an alternative method that is much more memory and resource efficient. The first approach employs a good old-fashioned machine learning (GOML) pipeline consisting of a tf-idf vectorizer with a logistic regression classifier, whose representative features are used to extract relevant highlights. The second, more resource intensive, uses an LLM for generating the summaries and is guided by chain-of-thought to provide sequences of text indicating clinical markers.

## 1 Introduction

Suicidal-themed messages on social media platforms can represent an indicator of suffering and mental health issues. According to Harmer et al. (2022), 6% of individuals aged 18-25 responded affirmatively to the survey questions on suicide ideation. Interdisciplinary work on psychology and computational linguistics (Zirikly et al., 2019; Uban et al., 2022) uses statistical models to identify various risks based on the content of social media posts or based on multi-modal characteristics such as time of post, user gender and class (Yang et al., 2022). Gaining awareness of the risk of suicide is essential, as it allows state organizations to offer support to those in need, and consequently, preventive measures can be taken, potentially saving the lives of those contemplating suicide. Therefore, it may be beneficial from multiple perspectives to develop methods through which the presence of suicidal thoughts can be determined on the basis of text posts on social networks. However, as Rezapour (2023) suggests, relying solely on algorithmic methods can introduce biases, risks, and, ultimately, case-by-case analyses must be carried out by experts.

In this paper, as part of the shared task of the 2024 Workshop on Computational Linguistics and Clinical Psychology (Chim et al., 2024), we address the identification of suicidal evidence in users' posts on Reddit by extracting phrases, expressions, key-words, and various types of summaries that can explain such labels. The shared task has been framed from the perspective of large language models (LLMs) with a suggestive title in this sense: "Utilising LLMs for finding supporting evidence about an individual's suicide risk level". Although LLMs are the current standard in natural language processing (McCoy et al., 2023; Hosseini et al., 2024), deploying such models at scale can be prohibitively expensive, while the pre-trainig can often be resource- and data-intensive, making such models available only for well-resourced languages and large research laboratories.

We address this task from the perspective of finding solutions for fast inference, and propose two variants: 1) to create a straightforward and *cheap* (as in time-efficient) pipeline for training and identifying suicidal evidence and 2) to use prompting with quantized LLMs (Dettmers et al., 2023) executed locally on CPU. The former is based on traditional machine learning classification techniques consisting of a tf-idf vectorizer over word ngrams paired with a feature importance selection process from a linear logistic regression classifier.

Our results in the shared task show that a machine learning pipeline can achieve competitive evaluation scores (top 3 recall) by leveraging the risk assement annotations from the provided dataset (Shing et al., 2018; Zirikly et al., 2019). However, our best-performing model is a combination of LLMs used to generate good-quality summarizations and machine learning to detect highlights.

*Proceedings of the 9th Workshop on Computational Linguistics and Clinical Psychology (CLPsych 2024)*, pages 256–263
March 21, 2024 ©2024 Association for Computational Linguistics

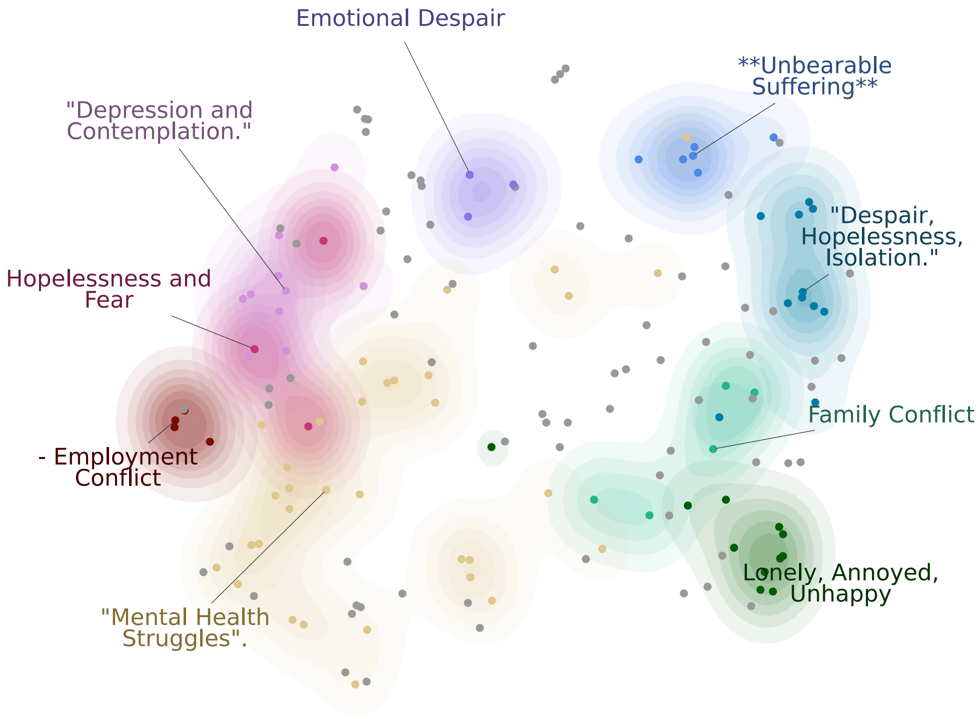

Figure 1: Major topics extracted from expert data labeled with openhermes-2.5-mistral-7b-q4_k_m.

## 2 Data Analysis

The annotated data provided for the shared task participants is identical to the previous edition CLPsych 2019 Shared Task: Predicting the Degree of Suicide Risk in Reddit Posts (Shing et al., 2018; Zirikly et al., 2019) and here we include a brief summary of its subdivisions: Task A: users on r/SuicideWatch Reddit annotated based on their risk level across multiple posts using crowd-sourced annotations. Expert: user posts annotated by experts of different specialties. Tasks B and C of annotations that we did not use in this work.

All data annotations contain suicide risk categories (Corbitt-Hall et al., 2016) marked with letters signifying different degrees: (a) no risk, (b) low, (c) moderate, and (d) severe risk. Expert data is of higher quality, it consists of 332 posts, the majority (49%) are labeled medium risk, followed by 28% high risk and 23% low risk. The 2024 Shared Task (Chim et al., 2024) evaluation data (not released to participants) contains additional annotations of suicide risk evidence (highlights and summaries) for 125 users of the expert subset. Our work only uses Task A and the expert subsets.

### 2.1 Topic Modelling

To have a first glance over the expert-annotated data, we use the BERTopic library (Grootendorst, 2022) and embed the documents with `BAAI/bge-small-en` a pre-trained English model (Xiao et al., 2023) which has the advantage of being relatively small and achieving good performance on the MTEB benchmark (Muennighoff et al., 2022). All document embeddings are projected into a bi-dimensional plane using a 5-neighbour UMAP (McInnes et al., 2018) configured to optimize the cosine similarity. The representations are clustered using HDBSCAN (McInnes et al., 2017) with a minimum cluster size of four. In a typical BERTopic pipeline, the topics are extracted using cTF-IDF and further fine-tuned using a representation model from openhermes-2.5-mistral-7b-q4_k_m[1]. The representation model is prompted with the following statement: *I have a topic that contains the following documents: [DOCUMENTS]. The topic is described by the following keywords: '[KEYWORDS]'. As an expert psychologist and therapist, provide a brief 5 word phrase to summarize the reason:.*

Figure 1 shows a result of this process with documents grouped by topic. Several key phrases are extracted using LLM prompts. Upon close inspection, the main topics in the dataset revolve around feelings of *despair*, *hopelessness*, socioeconomic hardships, and family conflicts. Our brief analyses

[1] `https://huggingface.co/TheBloke/OpenHermes-2.5-Mistral-7B-GGUF`

indicate that the texts contain strong signals for suicide and that very few subtleties can be observed in the assessment of risk degrees.

## 3 Good Old-fashioned Machine Learning (GOML)

The first approach, which also obtained the highest recall amongst submissions, is based on the following steps.

**1. Begin with Task A** crowd-annotated data and map the labels to binary, i.e., assigning the label 'a' to the value -1, and the labels 'b', 'c', and 'd' to the value +1. We fit a scikit learn logistic regression classifier on tf-idf features (Pedregosa et al., 2011). Tokenization is done using a regular expression of the form `r'\b[^\d\W]+\b'` and we employ a range of n-grams between 2 and 4 words. We cross-validate several models on different subsamples of risk annotations labeled as follows: **1.1 Test** - a model trained solely on Task A test set (186 posts), **1.2 TaskA** a model trained on the entire Task A, and **1.3 A+E** a model trained on both expert and TaskA data. Table 3 in the appendix contains the 5-fold cross-validation results that show relatively poor classifier performance.

**2. SHAP** SHapley Additive exPlanations (Lundberg and Lee, 2017) is an explainability library that implements several techniques to attribute individual contributions of each feature to a classifier's prediction. In our case, we use a simple linear explainer that assumes feature independence and ranks features based on a score computed as: $s_i = w_i(x_i - \hat{m_i})$, where $w_i$ is the classifier coefficient of feature $i$, $x_i$ is the feature value in a post and $\hat{m_i}$ the mean of the feature value across all posts.

**3. Selecting the highlights** requires matching the tokenized features from our tf-idf extractor to the text. We do so by aligning the different tokenizations using the Natural Language Toolkit (Bird et al., 2009) and retrieving the original verbatim strings. For highlight selection, we test **option 3.1** - highlights consisting of a context window of 14 words before and after each matched feature, not exceeding the sentence boundary. And **option 3.2** highlights consisting of entire sentences where important features are discovered in the original text.

**4. The summarization** consists of two options: **4.1** take the sentences found previously in step 3.2 and use an extractive summarization technique such as TextRank (Mihalcea and Tarau, 2004; Nathan, 2016) to generate a summary. This method is the fastest, but performed relatively poorly, obtaining high contradiction rates (0.238) and relatively low mean consistency (0.901). **Option 4.2 GOML+LLM** achieved the best overall performance and requires taking the sentences found previously and prompting a language model to generate an abstractive summary. Our best performing system in the official ranking is configured with option 3.2 (to extract full sentences as highlights) and option 4.2 (to generate summaries using LLM).

## 4 Language Models

For efficient text generation, we use a 4-bit quantized model (Q4_K_M) together with llama-cpp[2] and langchain (Chase, 2022) libraries. We use OpenHermes 2.5 based on Mistral (Jiang et al., 2023) that has been fine-tuned on code. According to the authors[3] training on a good ratio of code instruction of around 7-14% of the total dataset boosted several noncode benchmarks, including TruthfulQA, AGIEval, and GPT4All suite. The language models approach can be summarized in the following steps:

(a) prompt the model using langchain to extract highlights from the texts for a number of $K = 8$ times

(b) parse the LLM output and extract highlights from between quotation marks

(c) post-process responses: ensure the highlights are actually in the texts, remove duplicates, keep the longest matching highlights

(d) concatenate all posts and prompt the model without langchain to do a summary analysis of maximum 300 words

Text generation parameters are set to a temperature of 0.75, top-p nucleus sampling 1, and a maximum context size of 32000. To obtain as much data as possible, the LLM was run eight times on each post. The langchain prompt for extracting highlights is: *Provide sequences of text that indicate that this person is suicidal? \n \n Post Body: {post_body}*. Each response is saved and post-processed to extract valid highlights present in the text, to remove duplicates, and to preserve the longest matching highlight. The model tends to

[2] `https://github.com/ggerganov/llama.cpp`

[3] `https://huggingface.co/teknium/OpenHermes-2.5-Mistral-7B`

| submission | recall | precision | recall_w | harmonic |
|---|---|---|---|---|
| Test[1] | 0.921 | 0.888 | 0.513 | 0.904 |
| Test + LLM[2] | **0.939** | 0.890 | 0.390 | 0.914 |
| LLM[3] | 0.935 | **0.905** | **0.553** | **0.919** |
| TaskA_3.1 + LLM | 0.919 | 0.891 | 0.560 | 0.905 |
| A+E_3.1 + LLM | 0.918 | 0.892 | **0.578** | 0.905 |
| LLM duplicates | **0.941** | **0.907** | 0.398 | **0.924** |
| UoS NLP | 0.943 | **0.916** | 0.527 | **0.929** |
| sophiaADS | **0.944** | 0.906 | 0.489 | 0.924 |
| UZH_CLyp | 0.910 | **0.916** | **0.742** | 0.913 |

Table 1: Highlights evaluation scores of our systems in comparison to other participants in the Shared Task. The first three rows marked with superscript are the official versions we submitted during competition. The next 3 are additional experiments with highlights 3.1 or without removing duplicates and overlaps from LLM output. The last three rows are submissions from other participants.

| submission | consistency | contradiction |
|---|---|---|
| Test[1] | 0.901 | 0.238 |
| Test + LLM[2] | **0.973** | **0.081** |
| LLM[3] | 0.964 | 0.104 |
| TaskA_3.1 | 0.910 | 0.217 |
| A+E_3.1 | 0.908 | 0.218 |
| TaskA_3.1 + LLM | 0.971 | 0.085 |
| A+E_3.1 + LLM | **0.974** | **0.076** |
| UoS NLP | 0.966 | 0.107 |
| sophiaADS | 0.944 | 0.175 |
| UZH_CLyp | **0.979** | **0.064** |

Table 2: Summary evaluation scores of our systems in comparison to other participants in the Shared Task.

be more verbose, no matter how much we change the prompt, so the post-processing step proved to be essential.

To extract summaries, we run the model only once with the following prompt: *As a psychologist and expert therapist, summarize the content by identifying any indications of suicidal thoughts. Provide evidence from the text to support your analysis. \n \n Post Body: {content_body}\n \n Analysis:*. When using GOML with Option 4.2, the content body consists in the concatenation of important sentences instead of the post bodies. We found that the model tends to hallucinate and copy paste content from the text, unless the word *Analysis* is explicitly mentioned at the end.

## 5 Results and Discussion

Our three official submissions for the Shared Task in this order are:

- Test[1] - GOML fit on the Task A test set (1.1), highlights consisting of a 14 word context window (3.1), and extractive summaries generated from important sentences (4.1)
- Test + LLM[2] - [our best submission] GOML fit on the Task A test set (1.1), highlights consisting of entire sentences with important features (3.2), and LLM-generated abstractive summaries from combined sentences (4.2)
- LLM[3] - pipeline as described in section 4

Recall is computed as the average of the maximal semantic similarity between each gold highlight and all predicted highlights based on BERTScore (Zhang et al., 2019). A point of critique that we can raise here is that introducing duplicate highlights of different sizes will generate a better overall recall score. In practice, such a system could potentially slow down an expert looking for indicators of suicide. For example, our submission "LLM duplicates" from Table 1 does not remove highlights extracted from multiple runs of the LLM that are substrings of each other, and therefore obtains the highest recall. Similarly, systems that have shorter highlights (such as those that use the context around important features) achieve a lower recall than systems that return entire sentences as highlights. We do not know whether this is an artifact of BERTScore or from the way the annotations have been created. For example, the sophiaADS team (Tanaka and Fukazawa, 2024) returns complete sentences using a fine-tuned BERT model and their method obtains the highest recall score in the competition. In both their case and ours, we can observe that the weighted recall penalizes results in which highlights are entire sentences.

For this downstream task of identifying highlights, we did not observe significant improvements in performance when training the logistic regression classifier with more data, nor did we observe a degradation of performance when training on the smallest amount of samples consisting only of the test set of Task A. This is encouraging for potential extensions of the GOML methodology to less-resourced languages.

The generated summaries are evaluated by taking the probability scores (from an external NLI tool) of having a summary that contradicts the gold sentence as a premise. In terms of consistency and contradiction Table 2, the best results were obtained by Test + LLM[2] which combines the efficacy of extracting highlights of high recall (albeit low precision) with the ability of LLMs to generate adequate and coherent summary content. This is confirmed by the additional results combining LLM with GOML + option 3.1 with shorter summaries (Table 1 rows four and five). These models achieve the highest consistency (.974) and lowest contradiction scores (.076) of our systems. Team UZH_CLyp (Uluslu et al., 2024) uses retrieval augmented generation and provides additional context to the model when generating the summary to obtain the best results in the competition (given this criterion). This corroborates our observations that giving more concise or more focused content to LLMs leads to better generated summaries than providing the complete (and possibly noisy) post bodies from users to the LLM. The results of the team UoS NLP (Singh et al., 2024) are relatively similar to our LLM submissions that use chain-of-thought prompting to extract highlights and remove duplicates. Their LLM is based on Mixtral model quantized to 8 bits, which might explain the slight increase in evaluation scores across different metrics.

While GOML performs competitively to more resource-intensive approaches in detecting highlights, the same cannot be said about summaries. Our Test[1] model that used TextRank for extractive summarization obtained one of the worst contradiction and consistency scores in the entire competition. Its main advantage remains that it can run the entire machine learning pipeline to train the classifier and generate all the evidence (highlights and summaries) for the 125 users in less than 60 seconds. In contrast, our quantized LLM on CPU runs in 3.5 hours for the same set of users. To be consistent with our comparisons, in all of our approaches, we have only used a CPU server with 7 cores and 64 GB of memory to compute the results.

Given the surprising efficacy of the traditional machine learning model, we ask whether sentences containing important features have specific linguistic characteristics. Sentences are divided into two categories: **important** if they contain important features for classification and with the label **other** otherwise. Our statistical analyses visible also in Figure 2 indicate that important sentences are generally more likely to have pronouns, verbs, and adjectives. In terms of mean value, pronouns and verbs are statistically different at a p-value $< 0.05$ in important sentences more often than in the rest. Similarly, mean sentence lengths are statistically larger in important sentences than in the other ones. Adverbs show no difference between the two classes, and adjectives and nouns obtain a p-value of $0.6$ after 100,000 permutations. Given the nature of permutation tests, this is equivalent to saying that there is a 6% chance of observing a difference in means for adjectives and nouns due to chance.

Our brief analyses show that important sentences have different (statistically significant) linguistic patterns that can distinguish them from the rest. We believe that this could be one of the reasons behind the good evaluation scores and the suitability of the GOML approach to extract highlights from this particular dataset.

## 6 Conclusions

To conclude, our results show that a classifier paired with a machine learning explainability method can be a useful tool for identifying important sentences, phrases, and highlights that are representative of a given class. This is encouraging for languages where current LLMs do not perform as well or where the amount of data and compute resources is limited. Additionally, our experiments show that noisy generated output containing duplicates achieves better recall, leading to the conclusion that relying on a single metric can be detrimental to this task. We believe that ultimately expert human judgments would be the best measure for evaluating and selecting the most useful systems based on multiple criteria.

In general, when investigating the output of LLM-based approaches, we could observe better quality in terms of the generated text and langchain reasoning. Our work shows that these results can be further improved by combining LLMs with good old-fashioned machine learning methods.

## 7 Ethics

Working with user posts that talk about inflicting self-harm is a difficult endeavor. Although our methods bring about a small contribution in the interdisciplinary field of suicidology, we must recognize that technological solutions are not always helpful in an impactful way for people who suffer. Our work was carried out with the greatest care for the privacy and management of this data. During human analyses, repeated exposure to suicide-related content can be triggering and potentially harmful. The authors have double-checked each other on their mental health and ability to work during the entire time of doing this work.

## 8 Limitations

- Preserving duplicates or generating too many highlights can lead to an artificial increase in recall. The score increase can be misleading, since such a system can generate duplicates that are hard to interpret and not user-friendly.
- LLM-generated summaries may include sexist biases, we have not observed these in a systematic manner, but on occasion the LLM would assign gendered pronouns to users who did not explicitly mention this in their posts. Further research is required to integrate multi-modal variables such as class, race, gender in the prediction mechanism.
- The data that we have to work with had strong signals of suicide risk, therefore, we wonder whether such an approach would still be suitable in cases where the linguistic signal is more subtle or whether our models are able to generalize on out-of-domain data.

## Acknowledgements

We acknowledge the assistance of the American Association of Suicidology in making the dataset available. This work is supported by the Faculty of Mathematics and Computer Science, University of Bucharest, as part of the Archaeology of Intelligent Machines course.

## References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Harrison Chase. 2022. LangChain. Software. Released on 2022-10-17.

Jenny Chim, Adam Tsakalidis, Dimitris Gkoumas, Dana Atzil-Slonim, Yaakov Ophir, Ayah Zirikly, Philip Resnik, and Maria Liakata. 2024. Overview of the CLPsych 2024 Shared Task: Leveraging Large Language Models to Identify Evidence of Suicidality Risk in Online Posts. In *Proceedings of the Ninth Workshop on Computational Linguistics and Clinical Psychology*. Association for Computational Linguistics.

Darcy J Corbitt-Hall, Jami M Gauthier, Margaret T Davis, and Tracy K Witte. 2016. College students' responses to suicidal content on social networking sites: An examination using a simulated facebook newsfeed. *Suicide and Life-Threatening Behavior*, 46(5):609–624.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Bonnie Harmer, Sarah Lee, Abdolreza Saadabadi, et al. 2022. Suicidal Ideation. In *StatPearls [Internet]*. StatPearls Publishing.

Eghbal A. Hosseini, Martin Schrimpf, Yian Zhang, Samuel R. Bowman, Noga Zaslavsky, and Evelina Fedorenko. 2024. Artificial neural network language models predict human brain responses to language even after a developmentally realistic amount of training. *Neurobiology of Language*, pages 1–50.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *ArXiv*, abs/2310.06825.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

R Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *Transactions of the Association for Computational Linguistics*, 11:652–670.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11).

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive Text Embedding Benchmark. *arXiv preprint arXiv:2210.07316*.

Paco Nathan. 2016. PyTextRank, a Python implementation of TextRank for phrase extraction and summarization of text documents.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Mahdi Rezapour. 2023. Contextual evaluation of suicide-related posts. *Humanities and Social Sciences Communications*, 10(1):1–10.

Han-Chin Shing, Suraj Nair, Ayah Zirikly, Meir Friedenberg, Hal Daumé III, and Philip Resnik. 2018. "Expert, Crowdsourced, and Machine Assessment of Suicide Risk via Online Postings". In *"Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic"*, pages 25–36, New Orleans, LA. Association for Computational Linguistics.

Loitongbam Gyanendro Singh, Junyu Mao, Rudra Mutalik, and Stuart E Middleton. 2024. Extraction and Summarization of Suicidal Ideation Evidence in Social Media Content Using Large Language Models. In *Proceedings of the Ninth Workshop on Computational Linguistics and Clinical Psychology*. Association for Computational Linguistics.

Rika Tanaka and Yusuke Fukazawa. 2024. Integrating Supervised Extractive and Generative Language Models for Suicide Risk Evidence Summarization. In *Proceedings of the Ninth Workshop on Computational Linguistics and Clinical Psychology*. Association for Computational Linguistics.

Ana-Sabina Uban, Berta Chulvi, and Paolo Rosso. 2022. Explainability of depression detection on social media: From deep learning models to psychological interpretations and multimodality. In *Early Detection of Mental Health Disorders by Social Media Monitoring: The First Five Years of the eRisk Project*, pages 289–320. Springer.

Ahmet Yavuz Uluslu, Andrianos Michail, and Simon Clematide. 2024. Utilizing Large Language Models to Identify Evidence of Suicidality Risk through Analysis of Emotionally Charged Posts. In *Proceedings of the Ninth Workshop on Computational Linguistics and Clinical Psychology*. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Bing Xiang Yang, Pan Chen, Xin Yi Li, Fang Yang, Zhisheng Huang, Guanghui Fu, Dan Luo, Xiao Qin Wang, Wentian Li, Li Wen, et al. 2022. Characteristics of high suicide risk messages from users of a social network—sina weibo “tree hole”. *Frontiers in psychiatry*, 13:789504.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ayah Zirikly, Philip Resnik, Özlem Uzuner, and Kristy Hollingshead. 2019. "CLPsych 2019 shared task: Predicting the degree of suicide risk in Reddit posts". In *"Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology"*, pages 24–33, Minneapolis, Minnesota. Association for Computational Linguistics.

## A Appendix

The first text classification training scenario involves only the test set from Task A, because it is the smallest (186 posts), and one should expect it to generate the weakest classifier. We gradually increase the data to see whether there are changes in the results by adding the entire Task A data (labeled in the results section as "TaskA"). Lastly, we include the entire Task A and expert data, referred to in the results section as "A+E".

When running the tf-idf vectorizer we set the minimum document frequency to one, no limit on maximum features, Unicode strip accents, minimum number of documents set to one, enable the use of inverse document frequency (IDF) reweighting, smoothing to the IDF weights, and sublinear scaling to term frequency.

Logistic regression is set with balanced class weight, and we do not perform any hyperparameter optimization. Nevertheless, classifiers tend to predict only the majority class Table 3, so the balanced accuracy score never increases significantly, regardless of the fold or amount of data used.

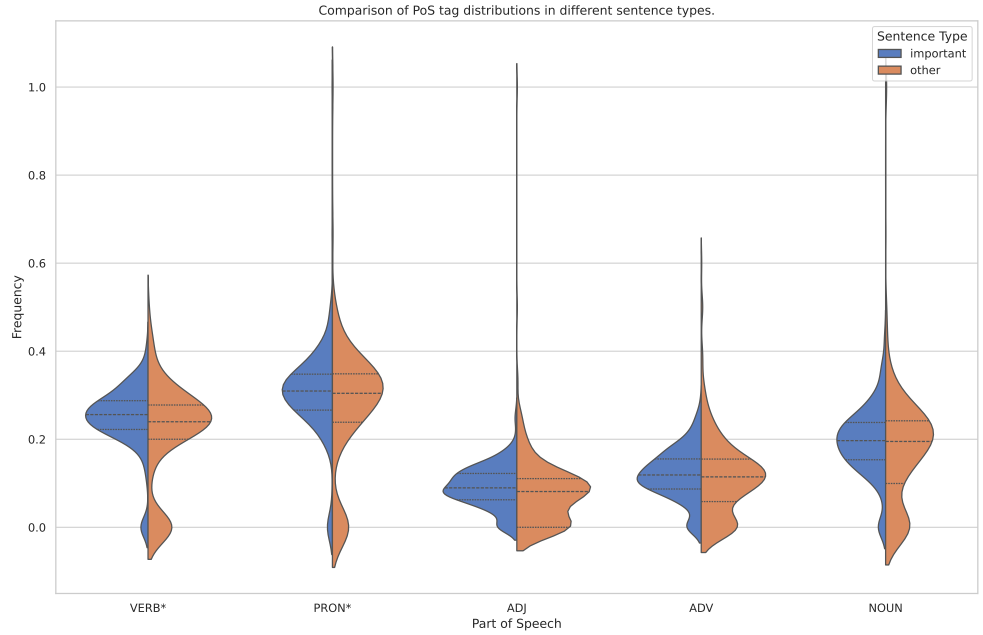

Figure 2: PoS tag distributions in sentences containing highlighted features (important) vs. other. Marked with * are PoS tags that have statistically significant means in a bootstrap permutation test at a p-value of $0.05$.

| Approach | Bal. Acc | Acc | F1 |
|---|---|---|---|
| test → Test | .5 | .82 | .74 |
| +train → TaskA | .5 | .82 | .74 |
| +expert → A+E | .5 | .86 | .8 |

Table 3: Stratified 5-fold cross-validation for binary risk prediction on different subsets of Task A and expert data. The first row represents cross-validation only on the test set, the second row adds the training set over the test set thus using the entire Task A, and the third row adds the expert data over all the previous. All values can vary between $\pm.05$ at different random shuffles.

## B What did Not Work

- Fine-tuning a LLM for classification with LoRA and unsloth library [4] using mistral-7b-bnb-4bit quantized model to classify the suicide risk by responding verbally; we were hoping to guide the model's attention towards important features for generating the content; after fine-tuning, the model was not able to produce good highlights.

- Given that OpenHermes 2.5 is fine-tuned on code, we were expecting to use grammars[5] to constrain the generation of highlights in the form of a list of strings, but the model proved not to perform very well in some of our empirical small-scale tests and we eventually abandoned this direction.

- We also tried to use Yake (Campos et al., 2020) to extract keywords from the titles and posts and then use this list of words as a parameter in TF-IDF. This approach did not work well because the list of extracted important features was too limited.

[4] https://github.com/unslothai/unsloth

[5] https://github.com/ggerganov/llama.cpp/blob/master/grammars