Key Findings

"What was the victim's story that led to their death?" It is hard to answer this question with the existing NVDRS coding guide, as events are often stored in boolean variables and limited to events that happened 1 month ago, thus, information about recurrences of events and the chronological order of events are lost. We recommend adding **temporal information** to events and allow for time-series analysis from the free-text narratives. Below are the top 3 interesting findings drawing from our analysis:

- **20%** of victims had repeated relationship problem with partner events occur prior to suicide. The median time between the 2 problem events is **2.5 days**, and the median time between the last relationship problem to death is **60 days**.
- Suicide attempts snowball into more problems: another suicide attempt (28% / 7 days), relationship problem with partner (17% / 3 days), arguments with family (3% / 60 days), buying a weapon (6% / 14 days)
- The most frequent last events prior to death were a prior suicide attempt (936 victims / **42 days median**), relationship problem with partner (497 victims / **14 days median**) and depressed mood or mental health (272 victims / **30 days median**).

Methodology

How did you decide what area related to youth mental health to explore?

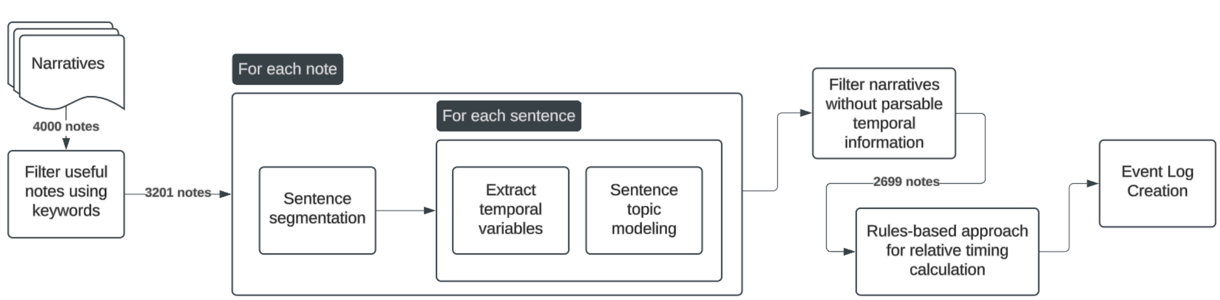
Patient stories are rarely documented as part of the patient chart. As one research study puts it: "The stories of people who attempt suicide are insufficiently reflected in suicide research in psychology" (Rimkeviciene 2016). In fact, according to many studies, including the NVDRS coding guide, many studies focused on only the most recent events that happened at most one month ago. Two excerpts of recent research papers are shown:

> "About half of people who die by suicide visit their primary care provider (PCP) within 1 month of doing so, compared with fewer than 1 in 5 contacting specialty mental health." (Dueweke 2018)

> "...suicide-related outcomes within one-week or one-month in individuals with current suicidal ideation (SI) or a recent suicide attempt (SA)." (Lengvenyte 2021)

To better guide suicide prevention, we must first be informed of the series of events that victims gone through days, weeks or even months prior to death. We hope to inspire future research to look into the patient story with a broader timeframe as well as population timeline trends to more effectively prevent suicides.

High Level Summary



Step 0 - Remove notes without temporal information. Not all narratives were used in this research. In the current dataset of 4000 narratives, we kept **3201** narratives that contained potential temporal variables. We used simple string matching on common temporal representations, such as "month", "day", "ago", etc.

Step 1 - Running valid notes through many rounds of Flan T5

- **Temporal Extraction**: Sequential Q&A through 3 flan-t5-xl prompts to segment the narrative into sentences and construct a structured temporal concept like `{'number': '1', 'unit': 'hour', 'before_or_after': 'before'}`
- **Sentence Topic Modeling**: We classified sentences into predetermined categories based on the existing boolean variables. For example, "V had just broken up with his girlfriend…" is classified as "relationship problem with partner" (`IntimatePartnerProblem`).

Step 2 - Event Log Creation

- **Filter narratives without temporal information**: 502 notes did not contain valid information or only contained the death with temporal variables. Those were removed from further processing. **2699** notes remained for final analysis.
- **Relative timing calculation heuristic function**: Each free-text temporal variable is now reformatted into an integer representing the relative hours prior to death. For example, "2 days ago" is reformatted into -48 (hours) and "2 months prior" is -1460 (hours)

The output of the process is an event log that can be used for data analysis purposes.

Final Result - Sankey Diagram from Event Log: We aggregated timelines per victim and constructed a Sankey diagram. Each node represents a significant event, such as relationship problem, suicide attempt. The transitions contain the median time to move to the next state, as well as the count of patients that moved to the next state. The last state is "Death of victim".

For development, We used flan-t5-large for inference locally, and flan-t5-xl for the final run. We opted to use the `Standard_NC24ads_A100_v4` on Azure (80GB vRAM) for speed purposes when running the final pipeline. The main Python libraries used are: `nltk` for sentence segmentation, `transformers` for T5 inference, and `plotly, matplotlib` for visualization.

Why did you decide to use these methods? How does your approach advance technical capabilities for studying youth mental health?

Creating a population journey requires an event log as input. By **turning free-text into a time-series format**, we unlock the data-rich sources which are previously unavailable to the regular data analyst and data scientist to perform downstream tasks such as determining a patient's most likely next step and present professionals with suitable interventions.

Pipeline Performance Evaluation

We used 7 hours to manually annotate 40 samples as the ground truth and compared them to the results generated by each step of our pipeline. Accuracy is calculated for all sentences if they are correctly extracted, but if there are false positives or missed sentences, it is calculated only for the correctly extracted ones.

Sentence extraction F1 score	Topic modeling accuracy	Temporal extraction accuracy
0.97	0.87	0.88

Are there any additional approaches you tried that did not make it into your final workflow (e.g., features, preprocessing steps, model types, etc.)?

We explored using large language models such as the llama-3.2 and Qwen2.5 families. Smaller models in the above families had questionable output accuracy, while large models were cost-prohibitive and time-consuming. GLiNER (Zaratiana 2023) for NER was also attempted but we ultimately landed on the flan-t5-xl (Chung 2024) as it provided the most promising results.

Furthermore, we decided that a Sankey diagram output is more visually understandable than running it through heuristic mining. In the end, we adhered to the "keep it simple" principle.

3 most impactful parts of our code

```python
def model_QA(query, question, timing_word):
    input_text = f"""
    question: {question}
    given sentence: {query}
    relative timing word: {timing_word}
    answer:
    """
    inputs = agent_tokenizer(input_text, return_tensors="pt")
    output_ids = agent_model.generate(
        inputs["input_ids"], max_length=500, num_beams=4,
        early_stopping=True,do_sample=True, temperature = 0.9
    )
    answer = agent_tokenizer.decode(output_ids[0], skip_special_tokens=True)
    return answer

def sequence_QA(query, questions, timing_word):
    answers = []
    for question in questions:
        answer = model_QA(query, question, timing_word)
        answers.append(answer)
    keys = ['number', 'unit', 'before_or_after']
    answer_dict = dict(zip(keys, answers))
    return answer_dict
```

Snippet 1: Sequence Q&A with Flan T5

Three prompts formed our temporal extraction step. Prompts are sequentially passed into a Flan T5 agent to produce our wanted output for each label, number, unit, and whether or not it occurred before or after death. Events that happened after death are discarded in post-processing.

Snippet 2: Relative timing calculation heuristic function

```python
if check_not_watch_time(query_num):
    query_unit = str(exact_timing['unit']).lower()
    query_order = str(exact_timing['before_or_after']).lower()
    query_category = str(category_ls[idx]).lower()

    if check_unit_valid(query_unit):
        if check_order_valid(query_order):
            if check_category_and_transform(query_category):
                query_unit = check_unit_valid(query_unit)
                query_order = exact_timing['before_or_after'] = check_order_valid(query_order)
                exact_timing['number'] = transform_to_hr(query_unit, query_num, query_order)
                exact_timing['unit'] = 'hour'
                category_ls[idx] = check_category_and_transform(query_category)

                tmp_useful_sentences.append(useful_sentences[idx])
                tmp_timing_words.append(timing_words[idx])
                tmp_category_ls.append(category_ls[idx])
                tmp_exact_timing.append(exact_timing)
```

We found that the most reliable way to format flan-t5-xl generated temporal concepts was with a rule-based method. This allowed us to quickly reorder events based on relative time, even if events were documented out of order. It also acts as an output control layer for T5 text2text generation.

Snippet 3: Grouping event log for Sankey visualization

From the event log, we grouped the same patient journeys together and aggregated based on the time to transition and the count. We do some processing to add the victim's death as the last state (as sometimes, the victim's death did not have associated temporal variables) and also transposed the times to prepare for Sankey visualization and easier calculation of the median.

```
import numpy as np

timeline_grouped_df = timeline_exclude_minutes_df.groupby("uid").agg(tuple).reset_index().groupby("category").agg(list).reset_index()
timeline_grouped_df["cnt"] = timeline_grouped_df.exact_timing.apply(len)
timeline_grouped_df.columns = ["journey", "uid", "times", "cnt"]
timeline_grouped_df["journey"] = timeline_grouped_df.apply(fix_journey, axis=1)
timeline_grouped_df["times_fixed"] = timeline_grouped_df.apply(fix_times, axis=1)
timeline_grouped_df["times"] = timeline_grouped_df["times_fixed"].apply(lambda x: np.transpose(x).tolist())
```

Other Takeaways and Next Steps

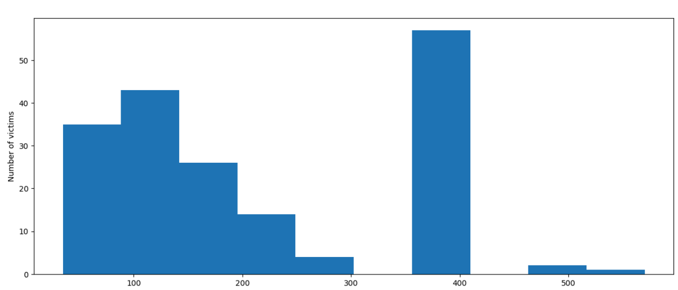
Ingesting more data: Only 2699 notes were possible to be analyzed in this research. Nonetheless, our pipeline supports more than just mental health narratives, as it was built to process any free-text clinical notes. Primary care and specialty care notes, emergency department discharge summaries, and even patient's personal diaries may allow for more detailed patient timelines.

More useful visualizations: Sankeys are one way to interpret time-series data in a human-readable format, but as our research outputs events in an event log format, the possibilities are endless. If more time was given, our next step would be to create a hidden Markov model based on existing data, popular in the clinical data science domain.

Visualizations

This Sankey diagram aggregates the most common timelines prior to victim death, scoped by journeys that were experienced by more than 30 victims and events that happened less than 1 hour to suicide or earlier than 2 years. As seen below, as the last observable event, more victims had attempted suicide 3 times than victims which had problems with their job. Relationship problems with partner is also the second most common last cause before the victim committed suicide, which was surprising to us.

Furthermore, many victims have attempted to commit suicide due to many reasons before their death, such as suicide ideations, alcohol, depressed mood or mental health.



This graph shows the time from suicide attempt to death for victims that had a suicide attempt of greater than 1 month. As seen, there is a significant increase of victims that commit suicide 1 year after a previous attempt. This is not a trend captured by existing research but is clear through data analysis.

References

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., ... & Wei, J. (2024). Scaling instruction-finetuned language models. Journal of Machine Learning Research, 25(70), 1-53.

Dueweke, A. R., & Bridges, A. J. (2018). Suicide interventions in primary care: A selective review of the evidence. Families, Systems, & Health, 36(3), 289–302. https://doi.org/10.1037/fsh0000349

Lengvenyte, A., Olié, E., Strumila, R., Navickas, A., Gonzalez Pinto, A., & Courtet, P. (2021). Immediate and short-term efficacy of suicide-targeted interventions in suicidal individuals: A systematic review. The World Journal of Biological Psychiatry, 22(9), 670–685. https://doi.org/10.1080/15622975.2021.1907712

Rimkeviciene, J., O'Gorman, J., Hawgood, J., & De Leo, D. (2016). Timelines for difficult times: use of visual timelines in interviewing suicide attempters. Qualitative Research in Psychology, 13(3), 231–245. https://doi.org/10.1080/14780887.2016.1170913

Zaratiana, U., Tomeh, N., Holat, P., & Charnois, T. (2023). Gliner: Generalist model for named entity recognition using bidirectional transformer. arXiv preprint arXiv:2311.08526.