Extracting Events from Nursing Notes: a MIMIC-III Case Study

Allmin Susaiyah and Natalia Sidorova

Eindhoven University of Technology, the Netherlands
{a.p.s.susaiyah,n.sidorova}@tue.nl

Abstract. Electronic health records (EHRs) contain rich clinical narratives critical for uncovering care patterns, identifying suboptimalities, and generating predictive insights. To enable data and process mining, unstructured text must be transformed into structured data. We propose a hybrid framework for extracting patient events from textual data that combines keyword matching, semantic sentence embeddings, and selective use of large language models (LLMs). LLMs are applied only to sentences where keyword and embedding-based methods disagree, reducing computational cost while maintaining extraction quality. Experiments show that this targeted use of LLMs improves clinical event detection efficiently.

Key words: clinical narratives, event logs, sentence embeddings, LLMs

1 Introduction

Clinical narratives, such as nursing notes, contain rich information that is often absent from structured electronic health record (EHR) data. These free-text entries capture observations about different aspects of patient's experience and care, such as daily life activities, social interactions, causal or temporal relationships (e.g. symptom relief following medication). Such details can be essential for generating meaningful insights into care processes that event logs derived from structured data alone cannot provide [4, 12].

Prior work has shown that converting free-text clinical narratives into structured records can reveal events and their relationships that were otherwise unknown [5, 10]. While event log extraction has been performed in other domains such as email, customer service conversations, customer relationship management activities, etc. [9], clinical narratives introduce unique challenges, including non-standardized vocabulary with the usage of frequent abbreviations, fragmented or grammatically incorrect sentences, and temporal ambiguity (e.g. a mix of past events and references to future or hypothetical events), which complicates the task of accurate event identification.

Table 1 illustrates some of these challenges using examples from the publicly available MIMIC-III [8] dataset. In these examples, some references to sleep indicate future events (will need), sleep of non-patients (family members), implicit events (not waking), and patient's wishes.

Table 1: Example sentences showing challenges in detecting *Sleep* events

Sentence	**Challenges**	***Sleep?***
SLEPT IN LONGS NAPS OVERNIGHT.	Omitted sentence subject	(Yes)
decrease due to sleepliness.	Typo, fragmented.	(No)
WELL THROUGH OUT NIGHT.	Ungrammatical; missing subject & predicate	(?)
NAPPING T/O AM.	Use of clinical shorthand	(Yes)
Will need bipap when sleeping.	Refers to a future (not current) event	(No)
Sister remained at bedside, sleeping on cot.	Irrelevant actor (not the patient)	(No)
patient not waking easily.	Implicit reference to sleep	(Yes)
asking for med. for sleep.	Related to future *Sleep* event	(No)
slept well but woke $\sim$ 2400 confused.	Provides context to above sentence	(Yes)

Traditional keyword matching (KM) approaches remain popular due to their low cost, transparency, and ease of implementation [16, 17, 18]. However, they lack the flexibility to handle variation or infer implicit mentions. Embedding similarity (EM)-based methods improve recall by capturing sentence-level semantic similarity to event type names [5, 11], but often struggle with precision and require careful calibration.

In recent years, large language models (LLMs) have shown great promise in information extraction tasks, including direct event extraction [7, 14] and data augmentation for training traditional models for event extraction [3]. However, their high computational cost and environmental impact raises concerns about their scalability and sustainability in clinical applications [13]. This motivates the need for more efficient, interpretable methods that can benefit from LLMs' semantic understanding while avoiding their high computational cost.

To reconcile these trade-offs, we propose a framework for extracting events from nursing notes adopting a hybrid approach: We start with applying KM and EM to detect candidates for clinically meaningful events in nursing notes and identify cases where the two methods disagree. We hypothesize that these cases are likely to be ambiguous or context-dependent. Then LLM-assisted event identification is performed on these sentences only, thus minimizing the LLM usage.

We hypothesize that *selective use of LLMs only on sentences where simpler methods (KM and EM) disagree leads to comparable overall extraction performance while significantly reducing computational cost.* To test this hypothesis, we (1) apply keyword and embedding-based models to a dataset extracted from the MIMIC-III dataset [8], (2) identify sentences with conflicting predictions, and (3) prompt an LLM to resolve these ambiguous cases. In doing so, we also explore a range of LLM prompt configurations that vary the informational content of the prompt and the requested output. Additionally, we use the LLM outputs to refine the keyword set, thereby improving the precision and recall of KM.

The paper is organized as follows. Section 2 introduces the hybrid event extraction approach. Section 3 describes the experimental dataset. Section 4 presents the results and analysis. Section 5 draws conclusions, and discusses the limitations and future work.

2 Hybrid Event Extraction Strategy

This section formalizes the task of clinical event extraction as addressed in this paper: identifying event types mentioned in individual sentences within a document. We define key concepts and describe hybrid approach combining three methods for event extraction: keyword matching (KM), embedding-based similarity (EM), and large language models (LLMs).

Event mention extraction Let Σ be the *vocabulary* of all tokens (words). A *sentence* $s \in \Sigma^+$ is a nonempty token sequence, and a *document* $d \in (\Sigma^+)^+$ is a nonempty sequence of sentences. We define a finite set $\texttt{EType} \subset \Sigma^+$ of *event types*, which are semantically meaningful phrases denoting clinical event categories (e.g. *Pain*, *Medication*, *Sleep*). A sentence may contain zero or more *event mentions*, e.g. "administered paracetamol", "patient complains about headache". The goal is to determine, for each sentence s, which event types from `EType` are mentioned in it.

Keyword matching Keyword matching [16, 17, 18] provides a low-cost baseline for event extraction. It requires no training, is highly interpretable, and can support initial deployment in clinical systems or bootstrap more advanced methods. Although unable to capture linguistic variation or implicit mentions, KM can still detects many explicit event mentions with minimal computational cost.

Each event type $e \in \texttt{EType}$ is associated with a set of keywords that are strong lexical indicators of the event. Keywords may be single words or phrases, e.g. *nap* (for event type *sleep*), or *bowel movement* (for event type *excretion*). Multiple keywords may map to the same event type, but each keyword is associated with only one event type. A sentence is assigned an event type if it contains all lemmas of a keyword phrase in the correct order and in any surface form (e.g. *nap*, *napped*, *napping*, all matching the lemma *nap*).

Embedding-based similarity Embedding-based matching (EM)[5, 11] goes beyond surface-level text patterns by leveraging the semantic content of the sentence and the event type name. Unlike KM, which relies on explicit lexical triggers and ignores the meaning of the event name, this method assumes that event type names carry meaningful information that can guide event detection. The use of pre-trained domain-specific models improves the results.[11]

To enable this, each sentence and event type are embedded into a shared vector space using a sentence encoder. The similarity $sim(s, e)$ between a sentence s and an event type $e \in \texttt{EType}$ is computed using cosine similarity of their corresponding embeddings. The event type e is assigned to the sentence s if the similarity $\text{sim}(s, e)$ exceeds a threshold θ, which can be determined empirically.

This method can capture paraphrased or implicit event mentions, for example, associating "not waking easily" with *Sleep*. However, similarity scores may be affected by unrelated content in longer sentences or ambiguity in short event names, requiring careful calibration.

General prompt template:
P = Given the sentence:`{sentence}`,
and the following event types:`{event_types_and_desc.}`,
choose the most relevant event type(s) for the sentence`{evd_out}`, `{evd_in}`
Output ONLY a JSON: `{"event type": <chosen event type>{extra_json}}`
Prompt variables:
Keyword evidences in the prompt(K_i)
`{evd_in}` = Additional facts: A keyword matching algorithm without context, detected keyword(s) $\mathcal{L}(l_j)$and assigned event type: e `{note}`
Embedder similarity evidences in the prompt(S_i)
`{evd_in}` = Additional facts: A sentence embedder, assigned the following similarity score to each of the event type labels: `sim(s,e)` for all $e \in E$ `{note}`
`{note}` = You may consider the additional facts if they are reasonable.Ignore otherwise
Keywords requested in the prompt (K_o)
`{evd_out}` = along with keywords that support each event type
`{extra_json}` = `, "keywords": {<chosen event type>: <keyword>}}`
Quotes requested in the prompt (Q_o)
`{evd_out}` = along with quotes from the text that support each event type
`{extra_json}` = `, "quotes": {<chosen event type>: <quote>}`

Fig. 1: LLM prompt template variants with options for input/output evidence.

Large Language Models Large language models (LLMs) have shown promising performance in natural language processing tasks, including text generation, question answering and sentiment analysis [2, 15]. Unlike KM or EM, LLMs can interpret nuanced phrasing, resolve ambiguity, and infer implicit mentions without explicit lexical cues.

In the LLM-based event extraction approach[7, 14], the task is framed as a prompt-based classification problem. The model is provided with a prompt P as shown in Figure 1 that includes the sentence s, a list of candidate event types $E = \{e_1, e_2, \ldots, e_k\}$, and a short task description. Optionally, the prompt may also include additional supporting information. The model generates a textual response, which is then parsed to identify the predicted event type(s).

LLM Configuration Space We explore various prompting strategies for LLM-based event extraction by modifying the content of the prompt and the format of the expected output. For each event type, we define a configuration space of 16 prompt variants, derived from four binary design choices: whether to include known event-specific **keywords in the input (K_i)**, whether to include embedding **similarity scores in the input (S_i)**, whether to ask the model to identify relevant **keywords in the output (K_o)**, and whether to ask the model to provide **quoted text as justification in the output (Q_o)**.

Hybrid method In this work, we investigate how overall performance and computational cost are affected when LLMs are used selectively, only on sentences

where KM and EM disagree. This strategy reserves computationally expensive LLMs for the most challenging cases, where their contextual understanding provides the greatest benefit.

We refer to sentences with conflicting results of KM and EM as *ambiguous* cases. For each ambiguous sentence, an LLM is prompted to detect relevant event types and identify evidence spans (e.g., keywords) using one of the 16 prompt configurations (cf. Fig 1).

The output of LLMs with the K_o = True is used for keyword set refinement: Sentences where the LLM does not detect an event, yet detects keywords, which are used to collect false positive triggers. Conversely, the LLM is used to propose new keyword candidates, which are manually reviewed to select those that are both accurate and novel. The refined set of keywords is used to update the keyword-matching rules, which are then re-applied to the dataset.

In our experiments, we assess how this workflow leverages the interpretability and speed of keyword-based and embedding-based methods while selectively integrating the semantic knowledge of LLMs without incurring unnecessary computational cost.

3 Data Selection, Preprocessing and Annotation

The experimental study focuses on identifying five event types in sentences from MIMIC-III [8] nursing notes: `EType` = {*Eating, Excretion, Sleep, Family, Pain*}, described in Table 2.

Data inclusion criteria The MIMIC-III dataset includes 1.85 million clinical notes from 61,293 hospital admissions of 46,520 patients. The majority (1.04M) are nursing-related notes. To construct a focused and representative subset, we apply the following inclusion and filtering steps:

- Note Category Filtering: Retain only "Nursing" or "Nursing/other" notes.
- Error Exclusion: Remove 374 reports known to contain patient mismatches.
- Patient Survival: Retain only patients who were discharged alive (n = 30,761).
- Temporal Coverage: For each patient, the number of reports is at least the number of hospital days.
- Age Filtering: Include only ages 18–89 to avoid unreliable or pediatric data.
- Length of Stay: Focus on patients hospitalized for 7–14 days. This range ensures sufficient data for temporal analysis and corresponds to the 80.4th percentile, capturing 42% of admissions.

Table 2: Description of event types and the number of sentences with keywords

Event Type	Description	Count
Eating	The patient takes food into their body by mouth.	1,902
Excretion	The patient discharges waste matter from their body.	5,991
Family	The patient has a visit, call, or communication with a family member.	8,152
Pain	The patient reports or shows signs of pain.	10,283
Sleep	The patient is sleeping, or the sleep's quality or quantity is described.	2,655

- Duplicate Timestamps: Exclude patients with reports sharing identical timestamps to avoid duplicated content.
- Report Density: Discard patients with more than five reports per day to mitigate reporting irregularities.
- After applying these criteria, we obtain a filtered dataset of 15,222 reports containing 338,395 sentences, referred to as the S-SET further on.

Data preprocessing We use the spaCy [6] for sentence segmentation, with additional rules to split at line breaks before section headers and at semicolons and perform basic cleaning (e.g., collapsing double spaces).

Keyword matching and embedding similarity For each event type $e \in \texttt{EType}$, we construct a KM-function by extracting synonyms, hypernyms, and hyponyms from WordNet [1], followed by manual filtering. Lemmas and their surface forms are compiled into keyword lists. Semantic similarity is computed using BioLORD [11], a biomedical sentence transformer.

Sentence sampling For each event type $e \in \texttt{EType}$, we construct a candidate sentence set F-SET, comprising both positive (keyword-matched) and negative sentences. We sample positives based on their similarity to the event type embedding, computed using BioLORD, drawing a fixed number of sentences per similarity bin. Negative samples include (1) sentences with keywords from other event types and (2) sentences with no event-related terms.

To separate reliable from ambiguous candidates, we estimate two event-specific similarity thresholds: θ^e_{pos} for keyword-matched sentences and θ^e_{neg} for keyword-free negatives. Each threshold is tuned using 3-fold cross-validation over the corresponding candidate set (S^e_{pos} or S^e_{neg}). For each sentence, we compute cosine similarity between its embedding and the embedding of event type e, using BioLORD. Within each fold, we search over a range of threshold values from 0.0 to 1.0 (step size = 0.01), selecting the one that maximizes F_1 score for classifying sentences based on keyword presence. The final threshold is obtained by averaging the best-performing thresholds across folds. These thresholds define three subsets within F-SET: the **Agreement Set (A-SET)**, which contains high-confidence positives and negatives where keyword and embedding signals agree; the **Disagreement Set (D-SET)**, which includes cases where the two signals disagree; and the **Threshold-Margin Set (M-SET)**, which consists of sentences confidently classified by embeddings only.

This partitioning supports both model evaluation and the prioritization of ambiguous examples for manual annotation. Note that D-SET and A-SET are complements of each other: A-SET $\cup$ D-SET = F-SET. The M-SET includes all instances from the D-SET along with additional high-agreement sentences from the F-SET: D-SET $\subset$ M-SET $\subset$ F-SET. For each event type, a test set (T-SET) is constructed by sampling from $S\text{-}SET \setminus F\text{-}SET$ to match F-SET's bin distributions, with fallback sampling from F-SET if needed.

Annotation For each event type e, the ground truth label G_e is 1 if the sentence mentions e (explicitly or implicitly), and 0 otherwise.

4 Experimental Results

In this section, we presents the evaluation of the hybrid approach for event detection in nursing notes, with a focus on the *Sleep* event type. While the proposed approach is applicable to all event types, we restrict our detailed analysis to *Sleep* due to the annotation effort involved in creating high-quality ground truth.

Only 8% of the 338,395 total sentences receive any event label under KM, and just 0.78% are labeled as *Sleep*, illustrating both the sparsity of target events and the heavy class imbalance. Precision is therefore emphasized when interpreting results, and evaluation strategies are designed to reflect this imbalance.

The number of sentences that were assigned an event type in S-SET using KM is shown in Table 2. For the *Sleep* event type, the most frequent keyword lemmas in the S-SET and F-SET subsets are: `sleep`: 1,369 (S-SET), 30 (F-SET), `sleeping`: 724 (S-SET), 30 (F-SET), `asleep`: 439 (S-SET), 30 (F-SET), `nap`: 267 (S-SET), 29 (F-SET), `bundle`: 15 (S-SET), 13 (F-SET).

4.1 Evaluation dataset construction

We follow the sampling procedure with the number of similarity bins $N_{\text{bin}} = 5$ and the manual annotation protocol, as described in Section 3, with samples stratified by keyword and lemma presence and embedding similarity score.

For positive examples, we sample at most $n_{\text{pos}}^{\text{Sleep}} = 6$ sentences per bin per sleep-related lemma from $S_{\text{pos}}^{\text{Sleep}}$, for negative examples, at most $n_{\text{neg}}^{\text{none}} = 6$ sentences per bin from $S_{\text{neg}}^{\text{Sleep}}$ that contain no event-specific keywords, and $n_{\text{neg}}^{\text{other}} = 1$ sentence per bin that contains a keyword from a different event type $e' \neq$ Sleep.

The performance metrics of the KM method for D-SET, M-SET, A-SET and F-SET are shown in Table 3. F-SET consists of 352 sentences: 131 containing *Sleep* keywords and 221 not. Manual annotation identified 112 positives and 240 negatives. On this set, KM achieves an F1 score of 0.905, with recall notably exceeding precision. The resulting thresholds used for prompting LLMs were $\theta_{\text{pos}}^{\text{Sleep}} = 0.09$ and $\theta_{\text{neg}}^{\text{Sleep}} = 0.48$. M-SET includes 192 sentences, comprising 57 positive and 135 negative examples. 65 sentences contain *Sleep*-related keywords and 127 sentences not. KM and EM disagree on about 10% of this sample.

By construction, KM and EM agree on all sentences in A-SET, 120 of which contain *Sleep*-related keywords and 213 not. Annotations label 108 of the sentences as mentioning *Sleep* and 225 not. Performance of KM drops significantly on the D-SET, which is by construction with 100% disagreement, with ambiguous or contradictory samples. This confirms the value of D-SET for testing LLM-assisted adjudication, particularly in edge cases.

Table 3: Performance of KM on different data subsets

Event type	**Metric**	**D-SET**	**M-SET**	**A-SET**	**F-SET**
Sleep	% Disagreement	100	10	0	5.3
	Precision	0.273	0.862	0.892	0.840
	Recall	0.750	0.982	0.991	0.982
	F_1 score	0.400	0.918	0.939	0.905

4.2 Performance of the LLM-based approach

We evaluated the LLM-based methods using the open-source Llama3 model [15] with 70B parameters and a temperature of 0. Table 4 presents the F_1 scores, precision and recall for various prompting strategies across D-SET, M-SET, and F-SET, with each row corresponding to a model variant described in Section 2. The best models are bolded; second-best are italicized. While these scores are influenced by the significant class imbalance in the datasets and should be interpreted cautiously, they provide useful comparative insights into model behavior.

As expected, KM comes as the best on F-SET, due to imbalance. However, on M-SET and D-SET, the LLM-based approaches outperform KM. M-SET contains more diverse and representative samples than D-SET, making it a better candidate for deeper analysis. Consequently, we focus on this set in the further analysis.

As shown in Table 4, providing ES (S_i) or requesting quotes in the output (K_o) does not improve performance. However, including the KM-detected keyword and event in the input (Ki) boosts both speed and F_1 score. Requesting both keyword and event names in the output (K_o) raises recall but sharply reduces processing speed.

Table 4: KW and LLM performance in identifying *Sleep* events.

Model	D-SET			M-SET			F-SET		
	F_1 score	Precision	Recall	F_1 score	Precision	Recall	F_1 score	Precision	Recall
KW Matching	0.400	*0.273*	**0.750**	0.918	0.862	**0.982**	0.905	0.840	**0.982**
LLM	0.333	0.250	*0.500*	0.915	0.885	0.947	0.857	0.832	0.884
$LLMK_i$	0.333	0.250	*0.500*	0.923	**0.900**	0.947	0.878	**0.856**	0.902
$LLMS_i$	0.333	0.250	*0.500*	0.915	0.885	0.947	0.849	0.802	0.902
$LLMS_iK_i$	0.182	0.143	0.250	0.915	0.885	0.947	0.842	0.789	0.902
$LLMK_o$	0.462	**0.333**	**0.750**	0.933	*0.889*	**0.982**	0.889	0.824	*0.964*
$LLMK_oK_i$	0.462	**0.333**	**0.750**	0.924	0.887	*0.965*	0.877	0.837	0.920
$LLMK_oS_i$	0.333	0.250	*0.500*	0.915	0.885	0.947	0.849	0.802	0.902
$LLMK_oS_iK_i$	0.182	0.143	0.250	0.915	0.885	0.947	0.842	0.789	0.902
$LLMQ_o$	0.333	0.250	*0.500*	0.915	0.885	0.947	0.831	*0.850*	0.812
$LLMQ_oK_i$	0.333	0.250	*0.500*	0.924	0.887	*0.965*	0.871	0.842	0.902
$LLMQ_oS_i$	0.333	0.250	*0.500*	0.915	0.885	0.947	0.849	0.802	0.902
$LLMQ_oS_iK_i$	0.182	0.143	0.250	0.915	0.885	0.947	0.842	0.789	0.902
$LLMQ_oK_o$	0.333	0.250	*0.500*	0.897	0.881	0.912	0.760	0.823	0.705
$LLMQ_oK_oK_i$	0.462	**0.333**	**0.750**	0.915	0.885	0.947	0.871	0.842	0.902
$LLMQ_oK_oS_i$	0.333	0.250	*0.500*	0.915	0.885	0.947	0.849	0.802	0.902
$LLMQ_oK_oS_iK_i$	0.182	0.143	0.250	0.915	0.885	0.947	0.842	0.789	0.902

Table 5: Inference Times and Confusion Matrices

I/p **O/p:**	LLM	$+K_o$	$+Q_o$	$+K_oQ_o$
LLM	4.199	12.565	14.823	23.715
$+K_i$	4.179	8.106	10.721	22.564
$+S_i$	4.293	4.214	4.380	4.252
$+K_iS_i$	4.482	4.337	4.490	4.318

(a) Average inference times (in seconds) of LLMs on F-SET.

		KM		$LLM\ K_i$		$LLM\ K_o$	
		1	0	1	0	1	0
GT	1	56	1	54	3	56	1
	0	9	126	6	129	7	128

(b) Confusion matrices on the M-SET for the task of *Sleep* mention detection

The effect of prompt design on the inference time on an A10 GPU is summarized in Table 5a. Prompt formats that require more output, such as K_o and Q_o, result in longer generation times. In contrast, supplying contextual evidence, such as keywords or similar sentences (K_i, S_i), tends to reduce latency and improve predictive performance. Including keyword evidence, in particular, improves both the generation speed and the F_1 score.

Table 5b compares the performance of KM and two LLM configurations selected for their simplicity in combination with high precision and recall scores: $LLMK_i$ and $LLMK_o$. The $LLMK_o$ model correctly identifies more true positives than $LLMK_i$ and more true negatives than KM, though at a significantly higher runtime. In contrast, the KM takes 20 ms per sentence, making it orders of magnitude faster than the LLMs. These results highlight the trade-off between speed and accuracy, and suggest that end-to-end event detection using LLMs remains impractical for large-scale deployment. This supports the hypothesis that LLMs are better suited for targeted event detection rather than full replacement for less computationally expensive techniques.

The structured outputs from LLMs, particularly in $LLMK_o$ and $LLMQ_o$, occasionally deviated from the expected JSON format. This caused partial information loss even when the content was semantically correct, underscoring the need for more robust output handling.

4.3 Qualitative error analysis

We qualitatively analyzed misclassifications made by the base model LLM, the keyword-input model ($LLMK_i$), and the keyword-output model ($LLMK_o$) on F-SET. While some differences in error types were observed between models, the sample sizes are too small to draw statistically significant conclusions. The results should be viewed as exploratory and indicative rather than conclusive.

LLMK$_i$ The $LLMK_i$ model showed promise in avoiding *false positives triggered by misleading keyword matches*. For example, both of the following sentences contain the word *sleep*, yet do not describe *Sleep* events:

`'asking for med. for sleep'`
`'the family also plans to sleep in the waiting room tonight'`

While the baseline LLM labeled these sentences as positive, LLM K_i correctly rejected them. This suggests that supplying keyword evidence in the prompt may help the model better contextualize the keyword and filter out semantically irrelevant matches.

Interestingly, $LLMK_i$ also avoided some *false positives on sentences without any keyword*, compared to the base LLM model. For example:

`'complaining of aching back pain from lying in bed on back'`
`'patient currently resting quietly in bed, reports comfortable and remains free of distress'`

These cases require the model to resist overgeneralizing from surface cues like "bed" or "rest". Improved performance may stem from explicit keyword evidence

in the prompt, helping the model better differentiate between relevant and irrelevant contexts overall, even when keywords are absent, resulting in more conservative predictions for sentences without keyword cues, thereby reducing false positives. However, further investigation is needed to confirm this hypothesis.

$LLMK_i$ correctly identified some indirect *Sleep* mentions that the base model missed. For instance:

`'..blood pressure down to 140's sleeping in short naps...'`
`'percocets started w improved relief,able to sleep in naps'`

These cases demonstrate the model's ability to link fragmented or colloquial descriptions to the *Sleep* concept, especially when guided by a relevant keyword ('nap' in this case). However, this ability was inconsistent: in another example with similar phrasing, $LLMK_i$ incorrectly classified the sentence as negative:

`'fentanyl 25mcg x 3 overnight for generalized discomfort, slept in long naps'`

This inconsistency suggests that the model's handling of phrases like *naps* requires further calibration.

LLMK$_o$ The $LLMK_o$ model gives fewer false negatives than the baseline, including cases requiring higher-level reasoning:

`'alert, confused most of night, oriented to self and often date, but often trying to get out of bed, looking for relatives, ...'`

Although this sentence lacks a Sleep keyword, the model correctly inferred poor *Sleep*. However, false positives arise from ambiguous contexts involving *rest* or *sedation*, which are lexically similar to *Sleep*:

`'patient is currently feeling a little better and resting in bed with family at bedside'`, `'patient is sedated on propofol and receiving morphine pro re nata for pain'`

In both examples, the model likely over-relied on cues like *resting* or *sedation*, which do not meet the clinical definition of *Sleep* in this task.

Overall, keyword-based prompting K_i shows promise to improve precision, especially in filtering spurious keyword matches and irrelevant behavior, while output keyword prompting K_o seems to improve recall, particularly in capturing implicit or indirect mentions, but may also lead to additional false positives.

4.4 Keyword refinement

To explore the potential of LLMs for keyword refinement, we examined nine sentences labeled as positive by KM but negative by LLM. Eight of these included the keyword *bundle* and were annotated as negative in the ground truth. The LLM correctly classified all eight, indicating that *bundle* is an unreliable keyword for *Sleep* and should be removed from the list of keywords.

To identify additional keyword candidates, we analyzed the structured outputs of the $LLMK_o$ model. Since the model occasionally deviated from the expected JSON format, we applied custom parsing techniques to extract candidate unigrams and bigrams. For the *Sleep* event, common unigrams included *night*,

rest, *bed*, *sleepy*, *fall*, and *overnight*, and bigrams, *throughout shift* and *wake easily*. Due to their general nature and low specificity, these were not incorporated into the keyword list.

However, *Excretion* event type yielded several high-precision candidates, like unigrams *urinal* and *diuresis*, and bigrams *lasix x*, *small inc*, *use urinal*, *void pass* and *pass stool*. Reliable terms were added to the keyword list. The keyword *discharge* were frequently flagged by the LLM as unrelated to *Excretion*, as it commonly indicated hospital discharge, and was removed from the keyword list.

After updating the keyword lists, the model was evaluated on T-SET. Improvements were observed: for *Sleep*, F_1 increased from 0.80 to 0.83 (McNemar's $p < 0.0002$); for *Excretion*, from 0.63 to 0.81 (McNemar's $p < 0.0139$).

4.5 Limitations

Several factors may have affected the outcomes. Clinical notes often lack proper punctuation and grammar, the sentences can be long or fragmented, compromising the results despite applying rule-based segmentation and the use of standard tools like SpaCy. Class imbalance, though accounted for in the analysis, limited statistical confidence for underrepresented event types, highlighting the need for larger, more balanced datasets. Some sentences are inherently ambiguous without additional context from other parts of the note or structured patient data. For example, "asking for med. for sleep" could reflect prior, current, or anticipated sleep issues, or unrelated intent.

We assumed a many-to-one keyword-to-event type mapping for interpretability and reproducibility. In practice, some keywords may map to multiple event types, making LLMs a promising direction for many-to-many associations.

5 Conclusion

We explored the potential of LLMs to enhance a simple, interpretable keyword matching (KM) approach for event extraction from clinical narratives, specifically nursing notes from the MIMIC-III dataset. Our experiments showed that the agreement and disagreement between KM and EM is a good indicator for nontrivial sentences on which LLM-based labeling pays off. LLMs proved useful for refining the keyword lists by suggesting additions and removals of keywords.

We compared several prompting strategies and found that incorporating keyword evidence into the input prompt enables the model to make faster and more reliable decisions. Additionally, prompting the model to identify and include keywords in its output further improves performance but takes significantly longer.

Future work will investigate how LLMs can support complex event log extraction including modeling temporal relationships and cross-sentence reasoning.

The code and the annotated dataset used in this paper are publicly available at `https://github.com/allmin/event_log_from_text`.

Acknowledgement: The research presented in this paper was funded by the NWO project TACTICS (628.011.004) and Lunet in the Netherlands.

References

1. Bond, F., Foster, R.: Linking and extending an open multilingual wordnet. In: Proc. 51st ACL (Long Papers). pp. 1352–1362 (2013)
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
3. Chen, R., Qin, C., Jiang, W., Choi, D.: Is a large language model a good annotator for event extraction? Proc. AAAI Conf. Artif. Intell. **38**(16), 17772–17780 (2024)
4. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **10**(3), e1346 (2020)
5. Geeganage, D.T.K., Wynn, M.T., ter Hofstede, A.H.: Text2el+: Expert guided event log enrichment using unstructured text. ACM Journal of Data and Information Quality **16**(1), 1–28 (2024)
6. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020)
7. Huang, F., Huang, Q., Zhao, Y., Qi, Z., Wang, B., Huang, Y., et al.: A three-stage framework for event-event relation extraction with large language model. In: Neural Information Processing. vol. 1968, pp. 434–446. Springer (2024)
8. Johnson, A., Pollard, T., Mark, R.: MIMIC-III clinical database (version 1.4). PhysioNet **10**(C2XW26), 2 (2016)
9. Kecht, C., Egger, A., Kratsch, W., Röglinger, M.: Event log construction from customer service conversations using natural language inference. In: 2021 3rd ICPM. pp. 144–151. IEEE (2021)
10. Korach, Z.T., Yang, J., Rossetti, S.C., Cato, K.D., Kang, M.J., Knaplund, C., et al.: Mining clinical phrases from nursing notes to discover risk factors of patient deterioration. Int. J. of Medical Informatics **135**, 104053 (2020)
11. Remy, F., Demuynck, K., Demeester, T.: BioLORD-2023: semantic textual representations fusing large language models and clinical knowledge graph insights. Journal of the American Medical Informatics Association **31**(9), 1844–1855 (2024)
12. Remy, S., Pufahl, L., Sachs, J.P., Böttinger, E., Weske, M.: Event Log Generation in a Health System: A Case Study. In: 18th BPM. LNCS, vol. 12168, pp. 505–522. Springer (2020)
13. Solovyeva, L., Weidmann, S., Castor, F.: AI-powered, but power-hungry? energy efficiency of LLM-generated code. In: 2025 IEEE/ACM 2nd Int. Conf. on AI Foundation Models and Software Engineering (Forge). pp. 49–60. IEEE (2025)
14. Susaiyah, A., Sidorova, N.: Zero-shot approaches for the extraction of event logs from medical notes. In: RCIS. LNBIP, vol. 547, pp. 435–451. Springer (2025)
15. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
16. Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., et al.: Clinical information extraction applications: a literature review. Journal of biomedical informatics **77**, 34–49 (2018)
17. Wu, P.H., Yu, A., Tsai, C.W., Koh, J.L., Kuo, C.C., Chen, A.L.: Keyword extraction and structuralization of medical reports. Health Information Science and Systems **8**(1), 18 (2020)
18. Xu, H., Stenner, S.P., Doan, S., Johnson, K.B., Waitman, L.R., Denny, J.C.: Medex: a medication information extraction system for clinical narratives. Journal of the American medical informatics Association **17**(1), 19–24 (2010)