

Supplementary Material

S1.1 Pseudocode: Hybrid Clinical Note Generation Pipeline

Pseudocode: Hybrid-2 (UMLS)

Input: Clinical note C
Output: Hybrid clinical notes H

- 1: **Preprocessing: Whitelist Vocabulary Construction-** $V_{permitted}$
- 2: Curate a vocabulary that is safe $V_{permitted}$
- 3: **De-identification**
- 4: $D \leftarrow \text{Deidentifier}(C)$ ▷ e.g., Philter, Flair, etc.
- 5: **Filtration (Retention and Redaction)**
- 6: $T_{struct} \leftarrow \{D \cap \{\text{Stopwords} \cup \text{Punctuation} \cup V_{permitted}\}\}$ ▷ NLTK Tokenizer
- 7: $T_{bio} \leftarrow \text{ExtractBiomedicalEntities}(D)$ ▷ e.g., Flair Biomedical NER
- 8: $T_{qty} \leftarrow \text{ExtractQuantities}(D)$
- 9: $T_{umls} \leftarrow \text{ExtractUMLS}(D)$ ▷ e.g., SciSpaCy umls_linker
- 10: $T_{union} \leftarrow \{T_{struct} \cup T_{bio} \cup T_{qty} \cup T_{umls}\}$
- 11: **Merging and Synthesis**
- 12: $T_{merged} \leftarrow \text{ResolveOverlaps}(T_{union})$ ▷ Sort by index, unify spans, mark omissions
- 13: $H \leftarrow \text{LLM-infer}(T_{merged})$ ▷ Instruct an LLM to infer the omissions
- 14: **return** H

Implementation Details

To ensure the reproducibility of the hybrid filtration framework, we detail the specific libraries utilized in our pipeline.

- Linguistic processing: Document segmentation and tokenization were performed using the NLTK `word.tokenize` module. To facilitate structural preservation, we utilized the NLTK `stopwords` corpus and the standard Python `string.punctuation` set as a primary filter for neutral linguistic markers.
- Named Entity Recognition (NER) and de-identification: Entity extraction was conducted using flair (v0.14.0) [1]. We utilized the `flair/ner-english-ontonotes-fast` model specifically to isolate the QUANTITY tag, ensuring the retention of clinical measurements while discarding potentially privacy sensitive entities such as PERSON, ORG, and GPE. Specialized biomedical extraction was handled by flair, retaining all identified terms. For comparison, we also used another de-identifier, Philter (v1.0.3) [2].
- UMLS terms extraction: Extraction was performed via `scispacy`, utilizing the `umls_linker` module. To ensure high precision in concept retention, we set the confidence threshold to $\tau = 0.98$.
- In this study, we created a candidate safe-word vocabulary utilizing NLTK’s Brown Corpus. The Brown Corpus contains roughly fifty thousand distinct words and it provides fine-grained part-of-speech tags, with a large tag-set that distinguishes,

for example, proper nouns, common nouns, cardinal numbers, and numerous specialized categories. This granularity allowed us to retain broad classes of likely non-sensitive tokens without relying on manual word-by-word curation while systematically excluding categories that are more likely to carry identifying content. We retained words occurring under 275 of the Brown Corpus’s 472 tags, including common nouns, pronouns, verbs, adjectives, adverbs, determiners, prepositions, conjunctions, and punctuation or structural markers. We excluded the remaining 197 tags, principally proper nouns and their variants, plural nouns and their variants, and cardinal or ordinal numbers. A word was admitted to the safe-word vocabulary only if it occurred exclusively under permitted tag categories in the corpus. Applying this procedure yielded a candidate safe-word vocabulary of 29,960 words. The occupation/job list was compiled from the publicly available Wikipedia “Lists of occupations” resource (https://en.wikipedia.org/wiki/Lists_of_occupations) and is now reproduced in the supplementary material.

Listing 1: Retained tags used for candidate safe-word vocabulary construction.

```
[", "''", '(', '(-HL', ')', ') -HL', '*-HL', '*-NC', '*-TL', ', , ', '-HL',
', -NC', ', -TL', '---', '---HL', ', , ', '-HL', '.-NC', '.-TL', ', , ', '-HL',
':-TL', 'ABN', 'ABN-HL', 'ABN-NC', 'ABN-TL', 'ABX', 'AP', 'AP$', 'AP+AP-NC',
', 'AP-NC', 'AT', 'AT-NC', 'AT-TL-HL', 'BE', 'BE-TL', 'BED*', 'BED-NC', '
BEDZ', 'BEDZ-HL', 'BEDZ-NC', 'BEM*', 'BEM-NC', 'BEN', 'BEN-TL', 'BER', 'BER
*', 'BER*-NC', 'BER-NC', 'BER-TL', 'BEZ', 'BEZ*', 'BEZ-NC', 'BEZ-TL', 'CC',
'CC-HL', 'CC-NC', 'CC-TL-HL', 'CD-NC', 'CS', 'CS-NC', 'CS-TL', 'DO', 'DO*-
HL', 'DO-HL', 'DO-NC', 'DO-TL', 'DOD', 'DOD*-TL', 'DOD-NC', 'DOZ*-TL', 'DOZ
-HL', 'DOZ-TL', 'DT', 'DT+BEZ', 'DT+BEZ-NC', 'DT+MD', 'DT-HL', 'DT-NC', 'DT
-TL', 'DTI', 'DTI-HL', 'DTI-TL', 'DTS', 'DTS+BEZ', 'DTS-HL', 'EX+BEZ', 'EX+
HVD', 'EX+HVZ', 'EX+MD', 'EX-HL', 'EX-NC', 'FW*-TL', 'FW+AT-HL', 'FW-BE',
'FW-IN+AT-TL', 'FW-IN+NN-TL', 'FW-JJT', 'FW-PPL', 'FW-QL', 'FW-UH', 'HV', '
HV*', 'HV-HL', 'HV-NC', 'HV-TL', 'HVD', 'HVD*', 'HVD-HL', 'HVG-HL', 'HVZ',
'HVZ*', 'HVZ-NC', 'HVZ-TL', 'IN', 'IN-NC', 'IN-TL', 'IN-TL-HL', 'JJ', 'JJ+
JJ-NC', 'JJ-NC', 'JJ-TL', 'JJ-TL-NC', 'JJR', 'JJR+CS', 'JJR-NC', 'JJR-TL',
'JJS-HL', 'JJT', 'JJT-NC', 'JJT-TL', 'MD', 'MD+HV', 'MD+TO', 'MD-NC', 'MD-
TL', 'NN', 'NN$', 'NN$-HL', 'NN$-TL', 'NN+BEZ-TL', 'NN+HVZ', 'NN+HVZ-TL', '
NN+NN-NC', 'NN-HL', 'NN-NC', 'NN-TL', 'NNS$-HL', 'NNS$-NC', 'NNS$-TL-HL', '
NR+MD', 'OD-NC', 'PN', 'PN$', 'PN+BEZ', 'PN+HVD', 'PN+HVZ', 'PN+MD', 'PN-HL',
', 'PN-NC', 'PN-TL', 'PP$', 'PP$-HL', 'PP$-NC', 'PP$-TL', 'PPL-HL', 'PPL-NC',
', 'PPL-TL', 'PPO', 'PPO-HL', 'PPO-NC', 'PPO-TL', 'PPS', 'PPS+BEZ-HL', 'PPS
+BEZ-NC', 'PPS+HVD', 'PPS+MD', 'PPS-HL', 'PPS-NC', 'PPS-TL', 'PPSS', 'PPSS+
BER-N', 'PPSS+BER-NC', 'PPSS+BER-TL', 'PPSS+BEZ', 'PPSS+HV', 'PPSS+HV-TL',
'PPSS+MD-NC', 'PPSS+VB', 'PPSS-HL', 'PPSS-NC', 'PPSS-TL', 'QL', 'QL-HL', '
QL-NC', 'QL-TL', 'RB', 'RB$', 'RB+BEZ', 'RB+BEZ-HL', 'RB+BEZ-NC', 'RB+CS', '
RB-NC', 'RB-TL', 'RBR', 'RBR-NC', 'RN', 'RP+IN', 'RP-NC', 'RP-TL', 'TO', '
TO-HL', 'TO-NC', 'TO-TL', 'UH', 'UH-HL', 'VB', 'VB+IN', 'VB+JJ-NC', 'VB+PPO',
', 'VB+RP', 'VB+TO', 'VB+VB-NC', 'VB-NC', 'VB-TL', 'VBD', 'VBD-NC', 'VBD-TL',
', 'VBG', 'VBG+TO', 'VBG-NC', 'VBG-TL', 'VBN', 'VBN+TO', 'VBN-HL', 'VBN-NC',
', 'VBZ', 'VBZ-NC', 'VBZ-TL', 'WDT', 'WDT+BER', 'WDT+BER+PP', 'WDT+BEZ', '
WDT+BEZ-HL', 'WDT+BEZ-NC', 'WDT+BEZ-TL', 'WDT+DO+PPS', 'WDT+DOD', 'WDT+HVZ',
', 'WDT-HL', 'WDT-NC', 'WPO-NC', 'WPO-TL', 'WPS+BEZ', 'WPS+BEZ-NC', 'WPS+
BEZ-TL', 'WPS+HVD', 'WPS+HVZ', 'WPS+MD', 'WPS-HL', 'WPS-NC', 'WPS-TL', 'WQL',
', 'WQL-TL', 'WRB', 'WRB+BER', 'WRB+BEZ', 'WRB+BEZ-TL', 'WRB+DOD', 'WRB+DOD
*', 'WRB+DOZ', 'WRB+MD', 'WRB-HL', 'WRB-NC', 'WRB-TL']
```

S1.2 LLM: Model, Parameters, and Prompt Design	093
We utilized a 4-bit quantized Llama-3.1-70B-Instruct model (AWQ) for filling the redacted gaps and performing fact extraction for evaluation [3]. To ensure the model effectively transitioned from a redacted gap-prompt to a coherent narrative, we employed a one-shot prompting strategy with the following sampling parameters and instructional design.	094 095 096 097 098 099
1. Sampling Configuration	100
The model’s decoding behavior was configured to allow for plausible clinical term invention. Temperature ($T = 0.7$) was Selected to allow for sufficient variability in inventing plausible terms. Top-p (0.9) was employed to ensure the model’s output remained within the most probable vocabulary. The model context length was 7000 tokens and the generation limit was set to 3300 tokens.	101 102 103 104 105
It is to be noted that, of the 1304 clinical notes in the dataset, only 21 documents exceeded the 3300-token limit. By capping the model at 7000 tokens, we prevented the <i>Out-of-Memory</i> (OOM) error. The 21 outlier notes that exceeded the 3300-token threshold were processed with truncation at the generation stage; however, given that these represented less than 1.7% of the total corpus, the impact on the aggregate semantic and privacy metrics remained negligible.	106 107 108 109 110 111
2. Prompt Template	112
The prompt was designed to cast the LLM as a medical assistant tasked with filling gaps with hypothetical data. The prompt structure is provided below:	113 114
<i>Instruction:</i> You are a medical AI assistant. The text below contains redacted gaps marked by [*]. Your goal is to reconstruct the note by replacing every [*] with plausible, hypothetical clinical information that fits the context. You must invent dates, names, and measurements to make the note complete. Do not preserve the [*] symbols. Return only the reconstructed note.	115 116 117 118 119
<i>One-Shot Example:</i>	120
Input: Patient [*] arrived at [*] complaining of [*].	121
Output: Patient John Doe arrived at 2:00 PM complaining of illness.	122
<i>Input Segment:</i>	123
Input: [Filtered Clinical Note]	124
Output:	125
3. Runtime	126
Due to usage restrictions on shared computation, the LLM-based note generation was performed using vLLM in a 4-way tensor-parallel configuration (<code>tensor_parallel_size=4</code> ; NVIDIA A100 40 GB each) with a maximum context length of 7,000 tokens, GPU memory utilization set to 0.50, and eager execution enabled. Under this configuration, one set of generation for Hybrid-2 (UMLS) took a total of 35 hours 20 minutes 38 seconds to complete. The de-identification and filtration stage was much quicker requiring approximately 12 minutes. The fact-extraction component used for evaluation took 16 hours 15 minutes 45 seconds. These timings provide a practical reference for the computational profile of the proposed framework under the reported configuration. Among the hybrid pipeline components, LLM generation was the most computationally intensive	127 128 129 130 131 132 133 134 135 136 137 138

stage, whereas de-identification was comparatively lightweight. The pipeline can process each note independently; so, generation can be parallelized and batched across inference workers; wall-clock time therefore decreases approximately proportionally with the number of available workers. In an applied deployment setting, this burden could be shifted from local multi-GPU inference to commercial APIs that support HIPAA-compliant use under an appropriate business associate agreement (BAA) or Healthcare Addendum. OpenAI, for example, states that eligible API services can support HIPAA-compliant use under a BAA. The generation cost can be moderate. For example, assuming 3,700 input tokens and 3,300 output tokens per note, standard processing of 1,304 notes would involve approximately 4.8 million input tokens and 4.30 million output tokens, corresponding to an estimated direct token cost of about USD \$3.3 ($4.8 \times \$0.15 + 4.3 \times \0.6) with `gpt-4o-mini` at current standard API rates. If batch processing is employed, the cost will be reduced by half.

S1.3 Prompt for Clinical Fact Extraction

Extract EXACT Substrings Only (Strict Mode)
You are given a clinical note. Extract atomic facts about the patient by copying text exactly as it appears in the note.

Hard rules

- **Exact substrings only:** Every item must be a verbatim substring of the note (same spelling, capitalization, punctuation, quotes, hyphens, abbreviations, spacing).
- **No paraphrasing or normalization:** Do not expand abbreviations (e.g., don't change "O2" to "oxygen"), don't fix typos, don't change tense, don't reorder words.
- Keep negation exactly as written.
- **Atomicity:** Each item must represent one clinical detail and be the shortest span that fully expresses that detail (don't merge multiple details).
- **Potential categories to consider (only if present as text):** Condition, Symptom, Finding, Medication, Procedure, History, Lab, Vital sign, Allergy, Demographic, or any other atomic clinical fact that appears verbatim.
- **Scope:** Use only facts about the patient; ignore provider credentials, addresses, and salutations unless they state a patient fact.

Mandatory self-check before finalizing

- For each item, confirm it appears verbatim in the note via an exact string match.
- If any character differs or the phrase isn't found exactly, remove or correct it.
- If your output contains multiple steps, only print the final step (i.e., comma-separated double-quoted list of facts).
- Wrap your final list with `<start>` and `<end>`.

Output format

- Output only a comma-separated list of facts.
- Wrap each fact in double quotes (this helps ensure exact copying and makes validation easy).
- Do not output anything else other than the comma-separated list of facts.

Here's the clinical note: {...clinical note text...}

S1.4 Output at different stages of clinical note processing

Table S1: Snippet showing different stages

Stage	Example
Raw	CC: 73 yo woman with h/o Sjogren's syndrome, Diabetes (?Type I), vascular insufficiency who now presents with syncopal episode. HPI: Ms Rivers was in her USOH until the morning of admission, when she went into the city to buy some buttons and pay a bill at ViewSonic . She noted that she was not feeling well throughout the morning, and she felt as if she may need to eat something. However, because she did not have her insulin and blood sugar monitoring equipment, she decided not to eat.
De-identified (Philter)	CC: 73 yo woman with h/o *****'s syndrome, Diabetes (?Type I), vascular insufficiency who now presents with syncopal episode. HPI: Ms ***** was in her USOH until the morning of admission, when she went into the city to buy some buttons and pay a ***** at ViewSonic . She noted that she was not feeling well throughout the morning, and she felt as if she may need to eat something. However, because she did not have her insulin and blood sugar monitoring equipment, she decided not to eat.
Synthetic note: Corresponding keywords and the generated note	Keywords: syndrome, Diabetes, Type, vascular insufficiency, presents with syncopal, episode, HPI, USOH, morning of admission, noted, morning, eat, insulin, blood, monitor, eat. Synthetic Note: Emily Wilson, a 67-year-old female with a history of Sjogren syndrome, Diabetes Type 2, and vascular insufficiency, presents with a syncopal episode. HPI: The patient was seen in the USOH clinic on the morning of admission and noted to have a syncopal episode in the morning after she attempted to eat.
Hybrid-1 note: Filtered note and the hybrid note	Filtered: [*] : [*] woman with [*] Sjogren's syndrome, Diabetes (? [*]), vascular insufficiency who now [*] with syncopal episode. [*] HPI: [*] was in her [*] until [*] of admission, when she went into the [*] to buy some [*] and [*] a [*] at [*] . [*] noted that she was not feeling well throughout [*] , and she felt as if she [*] need to eat something. [*] , because she did not have her insulin and blood sugar [*] equipment, she decided not to eat. Hybrid: 70 / female : 3 of Sjogren's syndrome, Diabetes (Type I), vascular insufficiency who now presents with syncopal episode. PMH: The patient was in her usual state of health until 10:00 of admission, when she went into the store to buy some groceries and experienced a dizzy spell at 13:45 . The patient noted that she was not feeling well throughout the day, and she felt as if she would need to eat something. However, because she did not have her insulin and blood sugar monitoring equipment, she decided not to eat.
Hybrid-2 note: Filtered note and the hybrid note	Filtered: CC: [*] woman with [*] / [*] Sjogren's syndrome, Diabetes (?Type I), vascular insufficiency who now [*] with syncopal episode. [*] HPI: [*] was in her [*] until the morning of admission, when she went into the city to buy some [*] and [*] a bill at [*] . [*] noted that she was not feeling well throughout the morning, and she felt as if she [*] need to eat something. [*] , because she did not have her insulin and blood sugar [*] equipment, she decided not to eat. Hybrid: CC: 72 year old woman with 35 / 40 Sjogren's syndrome, Diabetes (Type I), vascular insufficiency who now presents with syncopal episode. 72 HPI: She was in her apartment until the morning of admission, when she went into the city to buy some groceries and pay a bill at the bank . She noted that she was not feeling well throughout the morning, and she felt as if she would need to eat something. However , because she did not have her insulin and blood sugar monitoring equipment, she decided not to eat.

231 References

- 232
233 [1] Weber, L., Sanger, M., Munchmeyer, J., Habibi, M., Leser, U., Akbik, A.: Hun-
234 flair: an easy-to-use tool for state-of-the-art biomedical named entity recognition.
235 Bioinformatics **37**(17), 2792–2794 (2021)
- 236
237 [2] Norgeot, B., Muenzen, K., Peterson, T.A., Fan, X., Glicksberg, B.S., Schenk, G.,
238 Rutenberg, E., Oskotsky, B., Sirota, M., Yazdany, J., *et al.*: Protected health
239 information filter (philter): accurately and securely de-identifying free-text clinical
240 notes. NPJ digital medicine **3**(1), 57 (2020)
- 241
242 [3] Chung, P., Swaminathan, A., Goodell, A.J., Kim, Y., Reincke, S.M., Han, L.,
243 Deverett, B., Sadeghi, M.A., Ariss, A.-B., Ghanem, M., Seong, D., Lee, A.A.,
244 Coombes, C.E., Bradshaw, B., Sufian, M.A., Hong, H.J., Nguyen, T.P., Rasouli,
245 M.R., Kamra, K., Burbridge, M.A., McAvoy, J.C., Saffary, R., Ma, S.P., Dash,
246 D., Xie, J., Wang, E.Y., Schmiesing, C.A., Shah, N., Aghaeepour, N.: Verifact:
247 Verifying facts in llm-generated clinical text with electronic health records (2025)
248 [arXiv:2501.16672](https://arxiv.org/abs/2501.16672) [cs.AI]
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276