# Prompt-based Extraction of Social Determinants of Health Using Few-shot Learning

**Giridhar Kaushik Ramachandran**[1]*, **Yujuan Fu**[2]*, **Bin Han**[3]*
**Kevin Lybarger**[1], **Nicholas J Dobbins**[2], **Özlem Uzuner**[1], **Meliha Yetisgen**[2]

[1] Department of Information Sciences and Technology, George Mason University, Fairfax, VA, USA

[2] Department of Biomedical Informatics & Medical Education, University of Washington, Seattle, WA, USA

[3] Information School, University of Washington, Seattle, WA, USA

## Abstract

Social determinants of health (SDOH) documented in the electronic health record through unstructured text are increasingly being studied to understand how SDOH impacts patient health outcomes. In this work, we utilize the Social History Annotation Corpus (SHAC), a multi-institutional corpus of de-identified social history sections annotated for SDOH, including substance use, employment, and living status information. We explore the automatic extraction of SDOH information with SHAC in both standoff and inline annotation formats using GPT-4 in a one-shot prompting setting. We compare GPT-4 extraction performance with a high-performing supervised approach and perform thorough error analyses. Our prompt-based GPT-4 method achieved an overall 0.652 F1 on the SHAC test set, similar to the 7th best-performing system among all teams in the n2c2 challenge with SHAC.

## 1 Introduction and related work

Social determinants of health (SDOH) are the conditions in which people work and live that impact quality of life and health (Centers for Disease Control and Prevention, 2022). Understanding SDOH can assist in clinical decision-making (Daniel et al., 2018; Friedman and Banegas, 2018). SDOH is documented in the electronic health record (EHR) through unstructured clinical narratives and structured data; however, the clinical narrative includes a more detailed description of many SDOH events. To utilize the text-encoded SDOH information in secondary use applications, including clinical decision-support systems, the SDOH information must be automatically extracted (Daniel et al., 2018; Singh et al., 2017).

SDOH extraction has been explored using rule-based systems and data-driven models that use supervised learning (Hatef et al., 2019; Patra et al.,

2021; Yu et al., 2022; Han et al., 2022) on a variety of corpora (Uzuner et al., 2008; Stemerman et al., 2021; Yetisgen and Vanderwende, 2017). Recent SDOH extraction work utilizes large language models (LLMs) like BERT and T5, where models are fine-tuned to the SDOH extraction task (Lybarger et al., 2022; Romanowski et al., 2023). Recent advancements in LLMs, including larger models like Generative Pretrained Transformer (GPT)-based models (Brown et al., 2020; OpenAI, 2023), allow for new training paradigms, including few-shot or zero-shot learning. Recent developments in LLMs like the GPT-4 (OpenAI, 2023) and med-PaLM models have shown their capability to understand the clinical text and achieve/exceed human-level performance in US medical licensing exams (Singhal et al., 2022). This high performance may be attributed to (1) high model parameter counts, (2) large pre-training datasets, and (3) instruction tuning and optimization with Reinforcement Learning Human Feedback (RLHF) (Ouyang et al., 2022). Recent clinical information extraction (IE) work (Liu et al., 2023; Hu et al., 2023) comparing BERT-based fine-tuning approaches to zero-shot learning indicates GPT models can extract entities and relations with reasonable performance; however, there are many open questions related to the use of recent LLMs, like GPT-4, in clinical IE tasks.

In this work, we explore the extraction of SDOH using GPT-4 in a one-shot prompting setting with event-based SHAC (Lybarger et al., 2021). We compare prompt-based extraction approaches with a high-performing supervised BERT-based model(Lybarger et al., 2022) that has been fine-tuned to SDOH extraction from SHAC. We investigate two different one-shot prompting strategies for GPT-4, including prompts aimed at generating BRAT standoff format and inline annotations. We report an overall performance of 0.861 F1 from the fine-tuned model, evaluated on the withheld test set. The highest-performing one-shot GPT-4 approach

---

* Equal contribution

achieved an overall 0.652 F1 for SDOH event extraction. Our initial study shows that GPT-4 can extract SDOH information from text with limited training examples.

## 2 Data, Task, & Evaluation

The 2022 National NLP Clinical Challenges SDOH extraction task (n2c2/UW SDOH Challenge) used SHAC for model development and evaluation (Lybarger et al., 2023). SHAC contains 4405 de-identified social history sections of notes from MIMIC-III (Johnson et al., 2016) and the University of Washington (UW). SHAC includes training, development, and test partitions for both sources (MIMIC-III and UW). SHAC was annotated using BRAT (Stenetorp et al., 2012), a web-based annotation tool, to capture five SDOH event types: substance use (*Alcohol, Drug, Tobacco*), employment status (*Employment*), and living status (*LivingStatus*). Figure 1 (A) presents an annotated sample in BRAT from the SHAC UW training set.
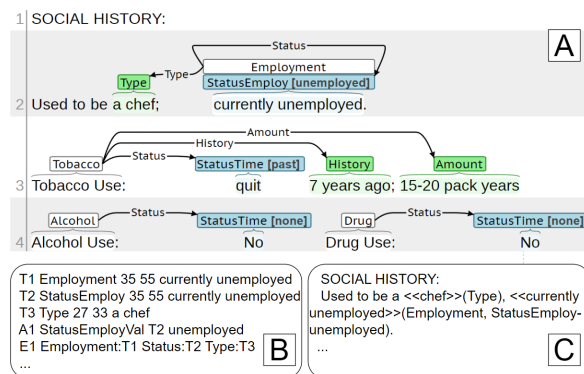


Figure 1: A. Sample note with SDOH events, visualized in the BRAT website. B. Standoff annotations in the BRAT format (.ann). C. Inline annotations.

The n2c2/UW SDOH Challenge evaluation criteria interpret the extraction as a slot-filling task (Lybarger et al., 2023). Each event comprises a single trigger span and at least one required argument. *Trigger any overlap* equivalence requires the predicted trigger to overlap with the true trigger of the same event type. Arguments can be classified into two categories: *span-only* (a multi-word span and argument type) and *labeled* (a multi-word span, argument type, and *subtype* label). Arguments can be equivalent only when attached to equivalent triggers. In addition to trigger *any overlap* equivalence, *span-only* argument equivalence is evaluated by *exact match*, and *labeled* arguments equivalence requires the correct argument and subtype labels

(*span agnostic*) (Lybarger et al., 2023).

We evaluated performance using the n2c2/UW SDOH Challenge criteria, as well as on more lenient evaluation criteria that still assess the clinical meaning of extraction. In the lenient criteria, trigger equivalence is relaxed to a minimum-distance metric (*minimum distance*), where gold triggers are paired (aligned) with the closest predicted trigger of the same event type, and the closest predicted trigger is counted as a true positive. In the lenient criteria, the *span-only* arguments use the *any overlap* criteria and the *labeled* arguments are evaluated as previously described.

## 3 Methods

We benchmark the SDOH extraction task using two methods: (1) multi-label variation of the Span-based Entity and Relation Transformer (SpERT)(Eberts and Ulges, 2020) architecture, mSpERT (Lybarger et al., 2022) benchmarked for SHAC as a high-performing fine-tuned baseline, and (2) prompt-based one-shot learning with GPT-4. Inspired by performance gains of few-shot learning, relative to zero-shot learning in prior work (Brown et al., 2020; Liévin et al., 2022), we use one-shot prompting with GPT-4 for the SDOH extraction task in this short study. We experiment with two distinct output formats - (1) BRAT-style standoff annotations (GPT-standoff) and (2) Inline annotations (GPT-inline).

We conduct the GPT-4 one-shot experiments through OpenAI's GPT-4 Chat Completion Application Programming Interface (API)[1], because of GPT-4's proprietary nature and significant hardware requirements. The API allows users to provide instructions via three role variables. Our prompts are structured in the following order:

1. ***system***: defines the desired role, personality traits, and task instructions for GPT-4. We use the *system* variable to assign GPT-4 the role of an annotator along with the paraphrased annotation guideline.

2. ***user***: provides an example note for one-shot learning.

3. ***assistant***: provides the gold annotations for the example note in *user*. This is an example prediction for one-shot learning.

Following the above definitions, we end with a *user* message containing a note to be annotated and

---

[1]https://platform.openai.com/docs/api-reference/chat

indicate the *assistant* should respond. We randomly sample a note from a subset of the SHAC-UW training set containing all five SDOH event types. Appendix A.2 contains formats and snippets of our prompts.

### 3.1 GPT-standoff

To assess GPT-4's ability to comprehend the task and generate structured outputs, we prompt the model to generate predictions in the BRAT standoff format (Stenetorp et al., 2012) used by SHAC. The BRAT standoff format includes pairs of text (*.txt) and annotation (*.ann) files. The event annotation is characterized by three BRAT annotation frames in the annotation file: (1) *Text bounds (T)* include a text span (e.g. "currently unemployed"), span label (e.g. 'Employment'), and character indices (e.g. "35 55") for marking both triggers and arguments; (2) *Attributes (A)* adds a subtype label to $T$, and (3) *Events (E)* characterize an SDOH event through linking a trigger and at least one argument. A visual representation is provided in Figure 1 (B).

We provide the paraphrased annotation guideline in the *system* and the note via *user* variables and elicit GPT-4 to output the desired annotation file through the *assistant* variable. Our preliminary experimentation indicated that though GPT-4 was able to correctly extract relevant text spans, it had some shortcomings: (1) some generated lines did not conform to the BRAT standoff format, and (2) the generated character indices did not correspond with the identified text spans. We post-processed the generated outputs to ensure compliance with the BRAT standoff format and updated the character indices to correspond with the first occurrence of the generated text span (<3% spans occur more than once). Please refer to Appendix A.3 for original and post-processed examples.

### 3.2 GPT-inline

Prior fine-tuned IE work utilized inline markers to infuse entity information in the body of narratives (Romanowski et al., 2023; Phan et al., 2021). In our work, we instruct GPT-4 to generate a version of the note with inline markers that identify the SDOH triggers and arguments. These markers encode all spans inside double-angle brackets ("« »"), with trigger, argument, and subtype labels appended (Figure 1, (C)). Similar to the GPT-standoff model, the GPT-inline model elicits GPT-4 the desired annotation format through the *assistant* variable. This method does not prompt the model to

make trigger-argument connections, and we use a heuristic search to associate each argument with the nearest event trigger, constrained by the allowable trigger-argument connections defined by the annotation guideline (See details in the Appendix. A.4). The GPT-inline output is post-processed into BRAT standoff format for evaluation.

## 4 Results and Discussion

Table 1 contains the overall performance of the prompt-based GPT-4 models on the withheld SHAC-UW test set, which was the evaluation data for Subtask C of n2c2/UW SDOH Challenge (Lybarger et al., 2023). The GPT-standoff and GPT-inline models achieve an overall F1 of 0.625 and 0.652, respectively. This performance is much lower than the mSpERT model and the highest-performing n2c2 systems, which utilized the entire training set in supervised model fine-tuning. GPT-inline achieved performance similar to the 7th best n2c2 system from IBM, which utilized BERT (Lybarger et al., 2023). This one-shot performance indicates that the natural language understanding capabilities of GPT-4 allow the prompt-based methods to leverage the annotation schema and achieve moderate performance. The results suggest that fine-tuning at least a portion of the training set may be needed to achieve high performance. We also observed that a generative architecture could lead to a new set of errors, and some of them may be eliminated in post-processing.

| Method | P | R | F1 |
|---|---|---|---|
| **Fine-tuned** | | | |
| Microsoft (T5) | 0.891 | 0.887 | 0.889 |
| CHOP (BERT) | 0.874 | 0.888 | 0.881 |
| mSpERT | 0.868 | 0.854 | 0.861 |
| ... | ... | ... | ... |
| IBM (BERT) | 0.538 | 0.788 | 0.640 |
| **GPT-4 one-shot + post-processing** | | | |
| GPT-standoff | 0.621 | 0.628 | 0.625 |
| GPT-inline | 0.650 | 0.654 | 0.652 |

Table 1: Comparison of overall micro-averaged performance of SDOH triggers and arguments between select top-performing models in the n2c2 challenge, mSpERT (fine-tuned baseline), and the GPT-4 models.

Table 2 contains trigger and argument micro-averaged F1 scores for each event type using the n2c2 and lenient evaluation criteria. Comparing overall performances (last row in the table), mSpERT outperforms both our GPT-4 models. But the performance gap between the fine-tuned model and the one-shot GPT models is smaller from the

| Field | Argument | # True Labels | n2c2 Evaluation (F1) | | | Lenient Evaluation (F1) | | |
|-------|----------|---------------|-------|-----------------|--------------|-------|-----------------|--------------|
| | | | mSpERT | GPT-standoff | GPT-inline | mSpERT | GPT-standoff | GPT-inline |
| **Trigger** | | | | | | | | |
| Alcohol | - | 403 | 0.964 | 0.861 | 0.938* | 0.967 | 0.972* | 0.952 |
| Drug | - | 473 | 0.929 | 0.824 | 0.861* | 0.942 | 0.935* | 0.898 |
| Tobacco | - | 434 | 0.963 | 0.825 | 0.917* | 0.970 | 0.965* | 0.939 |
| Employment | - | 153 | 0.908 | 0.803* | 0.709 | 0.915 | 0.921* | 0.766 |
| LivingStatus | - | 354 | 0.886 | 0.590 | 0.749* | 0.903 | 0.844* | 0.811 |
| **Labeled Argument** | | | | | | | | |
| Alcohol | StatusTime | 403 | 0.913 | 0.763 | 0.734 | 0.913 | 0.856* | 0.750 |
| Drug | StatusTime | 473 | 0.857 | 0.706* | 0.646 | 0.868 | 0.783* | 0.673 |
| Tobacco | StatusTime | 434 | 0.917 | 0.694 | 0.738* | 0.926 | 0.813* | 0.764 |
| Employment | StatusEmploy | 153 | 0.868 | 0.657 | 0.627 | 0.875 | 0.759* | 0.679 |
| LivingStatus | StatusTime | 354 | 0.833 | 0.572 | 0.709* | 0.850 | 0.787* | 0.760 |
| | TypeLiving | 354 | 0.871 | 0.560 | 0.725* | 0.891 | 0.770* | 0.774 |
| **Span-only Argument** | | | | | | | | |
| Alcohol | All types | 178 | 0.699 | 0.388* | 0.172 | 0.783 | 0.694* | 0.354 |
| Drug | All types | 418 | 0.625 | 0.219* | 0.104 | 0.688 | 0.426 | 0.381 |
| Tobacco | All types | 375 | 0.775 | 0.420* | 0.322 | 0.830 | 0.714* | 0.537 |
| Employment | Duration, History, Type | 96 | 0.675 | 0.169 | 0.109 | 0.735 | 0.677* | 0.500 |
| LivingStatus | Duration, History | 11 | 0.421 | 0.063 | 0.074 | 0.526 | 0.159 | 0.105 |
| | **Overall** | 5066 | 0.861 | 0.625 | 0.652* | 0.882 | 0.791* | 0.728 |

Table 2: Micro-averaged F1 comparison between mSpERT and GPT-4 one-shot models. SpERT outperforms the GPT-4 one-shot models in all trigger and argument extraction (with the exception of *Alcohol* and *Employment* triggers). For better readability, we only mark performance significance among GPT-4 one-shot models. * indicates performance significance among GPT-4 one-shot methods, with 10,000 bootstrap samples and a p-value threshold of 0.05.

n2c2 evaluation to the lenient evaluation, which can largely be attributed to higher trigger extraction performance, as argument equivalence requires trigger equivalence in both evaluations. The GPT-standoff model can identify the presence of SDOH events, but the identified triggers may not overlap with the gold trigger. The lenient evaluation only requires the same trigger type present in the social history text. The relatively lower performance for the GPT-inline model in *Employment* trigger extraction can be attributed to the model frequently identifying an *StatusEmploy* labeled argument without predicting an *Employment* trigger. The GPT-inline extractions may capture meaningful employment information but do not adhere to the annotation guidelines. For both GPT models, the extraction of *LivingStatus* triggers is relatively more challenging. Although the notes contain many plausible candidate spans for *LivingStatus* triggers, these spans were not annotated in SHAC since they did not contain information to resolve the associated *TypeLiving* labeled argument. The GPT models capture these false positive *LivingStatus* triggers often without *TypeLiving* labeled arguments. For argument extraction, GPT-standoff significantly outperforms GPT-inline in four arguments under the n2c2 evaluation and eight arguments in lenient evaluation. The GPT-inline

does not link annotated arguments to triggers, and a distance metric (character count) is used to link them, which contributes to the GPT-in-line's relatively lower performance. The labeled arguments are required for each event, and we observe that labeled argument performance is 0.1 F1 lower than the corresponding trigger performance. When multiple substance events are present in a note with differing *StatusTime* labels (e.g. current and past), we observe the GPT-standoff model tends to output the same *StatusTime* label for all substance events. The GPT-inline model correctly captures both substance trigger and *StatusTime* spans with the correct label but fails to correctly link the triggers with the right *StatusTime* spans because multiple *StatusTime* spans can have the same distance to a trigger span.

## 5 Conclusions

We investigate the efficacy of two prompt-based approaches for extracting SDOH from social history sections using GPT-4. Although the supervised model achieves higher performance, our findings indicate that GPT-4's one-shot learning capabilities serve as a promising starting point for extracting SDOH events without the need for annotated data. Possible gains in future work may be achieved with a combination of few-shot and active learning.

# 6 Limitations

We only explored one-shot prompting strategies with GPT-4. More examples (few-shot) may improve performance. We prompted GPT-4 with only a single randomly selected sample that included all of the annotated event types. Our post-processing included simple rules to process the generated output and may be improved. The quality of the sample and the selection method may influence performance. We explored two prompting styles. Future work could explore more prompting methods such as question & answering and chain-of-thought (Wei et al., 2023) and fine-tuning non-proprietary LLMs.

# 7 Ethics statement

Our experimentation utilized OpenAI API to extract SDOH information from SHAC with GPT-4. SHAC is a fully de-identified corpus of social history sections. The use of such external API/models could introduce ethical problems related to privacy, identifiability, and other unintended consequences if the data sets are not fully de-identified. Additionally, a careful examination is needed to assess potential bias in LLMs for extracting SDOH prior to implementing real-life secondary use applications. We received approval from the Institutional Review Board (IRB) prior to conducting the presented research. As our GPT-4 one-shot experiments are conducted on the SHAC-UW test set, broader use of the model may need necessary precautions.

# 8 Acknowledgements

# References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Centers for Disease Control and Prevention. 2022. Social determinants of health at CDC.

Hilary Daniel, Sue S Bornstein, and Gregory C Kane. 2018. Addressing social determinants to improve patient care and promote health equity: An american college of physicians position paper. *Annals of internal medicine*, 168(8):577–578.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *24th European Conference on Artificial Intelligence*.

Nicole L Friedman and Matthew P Banegas. 2018. Toward addressing social determinants of health: a health care system strategy. *The Permanente journal*, 22.

Sifei Han, Robert Zhang, Lingyun Shi, Russell Richie, Haixia Liu, Andrew Tseng, Wei Quan, Neal Ryan, David Brent, and Fuchiang (Rich) Tsui. 2022. Classifying social determinants of health from unstructured electronic health records using deep learning-based natural language processing. *Journal of Biomedical Informatics*, 127:103984.

Elham Hatef, Masoud Rouhizadeh, Iddrisu Tia, Elyse Lasser, Felicia Hill-Briggs, Jill Marsteller, and Hadi Kharrazi. 2019. Assessing the availability of data on social and behavioral determinants in structured and unstructured electronic health records: A retrospective analysis of a multilevel health care system. *Journal of Medical Internet Research*, 7.

Yan Hu, Iqra Ameer, Xu Zuo, Xueqing Peng, Yujia Zhou, Zehan Li, Yiming Li, Jianfu Li, Xiaoqian Jiang, and Hua Xu. 2023. Zero-shot Clinical Entity Recognition using ChatGPT. *arXiv preprint arXiv:2303.16416*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, et al. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:160035.

Valentin Liévin, Christoffer Egeberg Hother, and Ole Winther. 2022. Can large language models reason about medical questions? *arXiv preprint arXiv:2207.08143*.

Zhengliang Liu, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Wei Liu, Dinggang Shen, Quanzheng Li, et al. 2023. Deid-gpt: Zero-shot medical text de-identification by gpt-4. *arXiv preprint arXiv:2303.11032*.

Kevin Lybarger, Nicholas J Dobbins, Ritche Long, Angad Singh, Patrick Wedgeworth, Ozlem Uzuner, and Meliha Yetisgen. 2022. Leveraging Natural Language Processing to Augment Structured Social Determinants of Health Data in the Electronic Health Record. *arXiv preprint arXiv:2212.07538*.

Kevin Lybarger, Mari Ostendorf, and Meliha Yetisgen. 2021. Annotating social determinants of health using active learning, and characterizing determinants using neural event extraction. *Journal of biomedical informatics*, 113:103631.

Kevin Lybarger, Meliha Yetisgen, and Özlem Uzuner. 2023. The 2022 n2c2/uw shared task on extracting social determinants of health. *Journal of the American Medical Informatics Association*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Braja Patra, Mohit Sharma, Veer Vekaria, Olga Patterson, Benjamin Glicksberg, Lauren Lepow, Euijung Ryu, Joanna Biernacka, Al'ona Furmanchuk, Thomas George, William Hogan, Yonghui Wu, Xi Yang, Bian Jiang, Myrna Weissman, Jaya Priya, J. Mann, Mark Olfson, Thomas Jr, and Jyotishman Pathak. 2021. Extracting social determinants of health from electronic health records using natural language processing: A systematic review. *Journal of the American Medical Informatics Association*, pages 1–12.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature.

Brian Romanowski, Asma Ben Abacha, and Yadan Fan. 2023. Extracting social determinants of health from clinical note text with classification and sequence-to-sequence approaches. *Journal of the American Medical Informatics Association*. Ocad071.

Gopal K Singh, Gem P Daus, Michelle Allender, et al. 2017. Social determinants of health in the united states: addressing major health inequality trends for the nation, 1935-2016. *International journal of MCH and AIDS*, 6(2):139.

Karan Singhal, Shekoofeh Azizi, Tao Tu, et al. 2022. Large language models encode clinical knowledge.

Rachel Stemerman, Jaime Arguello, Jane Brice, Ashok Krishnamurthy, Mary Houston, and Rebecca Kitzmiller. 2021. Identification of social determinants of health using multi-label classification of electronic health record clinical notes. *American Medical Informatics Association, issuing body*, 4.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Ozlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane. 2008. Identifying patient smoking status from medical discharge records. *American Medical Informatics Association*, 15:14–24.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Meliha Yetisgen and Lucy Vanderwende. 2017. Automatic identification of substance abuse from social history in clinical text. In , pages 171–181.

Zehao Yu, Xi Yang, Chong Dang, Songzi Wu, Prakash Adekkanattu, Jyotishman Pathak, Thomas George, William Hogan, Yi Guo, Jiang Bian, and Yonghui Wu. 2022. A study of social and behavioral determinants of health in lung cancer patients using transformers-based natural language processing models. *AMIA Annual Symposium proceedings. AMIA Symposium*, 2021:1225–1233.

## A Appendix

### A.1 The SHAC annotation schema

| Event type | Argument type | Argument subtypes | Span examples |
|---|---|---|---|
| Alcohol, Drug, & Tobacco | StatusTime* | {none, current, past} | "denies," "no history" |
| | Amount | – | "1 pack", "3 drinks" |
| | Duration | – | "since last week" |
| | Frequency | – | "1-2x/week", "daily" |
| | History | – | "when he was young" |
| | Type | – | "smokeless," "methamphetamine" |
| Employment | StatusEmploy* | {employed, unemployed, retired, on disability, student, homemaker} | "works," "unemployed" |
| | Duration | – | "since last week" |
| | History | – | "10 years ago" |
| | Type | – | "remote office work" |
| LivingStatus | StatusTime* | {current, past, future} | "lives," "lived" |
| | TypeLiving* | {alone, with family, with others, homeless} | "with husband," "alone" |
| | Duration | – | "since he was 12" |
| | History | – | "until 2 years ago" |

Table 3: Annotation guideline summary. *indicates the argument is required

### A.2 Prompt Methods

The exact prompting messages are listed below. The annotation guideline is used for the SHAC dataset creation. We removed all the annotation examples in the guideline, as some of the examples are from MIMIC-III. We also remove invalid references to tables.

#### A.2.1 Message 1 - *System*

**GPT-4 role definition for standoff annotation**
You are an expert medical annotator and understand the BRAT standoff format very well. You are given a document that contains the following list of entities and events:"

**GPT-4 role definition for inline annotation**
"You are an expert medical annotator who adds annotations as inline markers in documents. You are given a document to annotate the following list of entities, events, and attributes:

**Annotation guideline**
The annotation involves the identification of SDOH events, where each SDOH event is represented by a trigger and set of entities. The trigger consists of a multi-word span (word or phrase) and a label indicating the type of SDOH (e.g. employment or tobacco use). All annotated phenomena are defined in terms of the span (words associated with phenomena) and the span type (e.g. amount, status, etc.), and some annotated phenomena, like status, will also include a span label (e.g. current or past). If an annotated span includes a noun, the selected span should include the entire applicable noun phrase. If an annotated span includes a verb, the selected span should include applicable adverbs. The inclusion of entire noun phrases and relevant adverbs is applicable to all annotated phenomena. You are given a document that contains the following list of events and entities:

[EVENTS]

- Alcohol: Alcohol event indicates the usage of alcohol. The trigger span should be a noun phrase describing a general substance type, like "alcohol" if present. Alcohol may be abbreviated ETOH

in chart notes. If general substance types are not present, the span should be a relevant verb (e.g. "drink") or a noun phrase describing a specific substance type (e.g. "beer"). Examples of Alcohol spans include: ["alcohol", "beer", "drink"]. ...

[ENTITIES]
<Span only entities>

- Amount: Linked to substance use events (Alcohol, Drug, Tobacco). Amount indicates the quantity of the substance used by the patient (e.g. "three drinks" or "2 packs") and do not include qualitative descriptions (e.g. "significant" or "heavy") . Amount spans should not include frequency information, and frequency and amount spans should not overlap. Examples of Amount entities include: ["one glass", "up to three drinks", "1 ppd", "half a pack", "4 packs"].

  ...

### A.2.2 Message 2 - *User*

**Annotation instructions for standoff annotation**
Based on this annotation guideline, please annotate the following document in the BRAT standoff format.

**Annotation instructions for inline annotation**
Based on this annotation guideline, please annotate the following document with inline markers.

**An example document**
HABITS:
Tobacco Use: denies Alcohol Use: denies Drug Use: denies

SOCIAL HISTORY:
Residence: [LOCATION] with husband and kids
Job: no longer works

### A.2.3 Message 3 - *Assistant*

**Annotated example in standoff annotation**
T1 LivingStatus 88 97 Residence
T2 TypeLiving 110 131 with husband and kids
T3 StatusTime 88 97 Residence
...
T11 StatusEmploy 137 152 no longer works
E1 LivingStatus:T1 Type:T2 Status:T3
E2 Tobacco:T4 Status:T6
...
A1 TypeLivingVal T2 with_family
A2 StatusTimeVal T3 current
...

**Annotated example for inline annotation**
HABITS:
«Tobacco Use»(Tobacco): «denies»(StatusTime-none) «Alcohol Use»(Alcoho): «denies»(StatusTime-none) «Drug Use»(Drug): «denies»(StatusTime-none)

SOCIAL HISTORY:
«Residence»(LivingStatus, StatusTime-current): [LOCATION] «with husband and kids»(TypeLiving-with_family)
«Job»(Employment): «no longer works»(StatusEmploy-unemployed)

### A.2.4 Message 4 - *User*

The annotation instructions for standoff and inline annotations are the same as in Message 2.

## A.3 Examples of Post-processing

The example document before and after post-processing can be found in Table. 4

| | Before Post-processing | After Post-processing |
|---|---|---|
| **GPT-standoff** | T1 LivingStatus 26 35 Residence<br>T2 TypeLiving 43 64 with husband and kids<br>T3 StatusTime 75 97 Residence<br><br>...<br>T11 StatusEmploy 110 123 no longer works<br>A1 TypeLivingVal T2 with_family<br>A2 StatusTimeVal T3 current<br><br>...<br>E1 LivingStatus:T1 Type:T14 Status:T3<br>... | T1 LivingStatus 88 97 Residence<br>T2 TypeLiving 110 131 with husband and kids<br>T3 StatusTime 88 97 Residence<br><br>...<br>T11 StatusEmploy 137 152 no longer works<br>A1 TypeLivingVal T2 with_family<br>A2 StatusTimeVal T3 current<br><br>...<br>E1 LivingStatus:T1 Status:T3<br>... |
| **GPT-inline** | ...<br>SOCIAL HISTORY:<br>«Residence»(LivingStatus, StatusTime-current): [LOCATION] «with husband and kids»(TypeLiving-with_family)<br>«Job»(Employment) Status: «no longer works»(StatusEmploy-unemployed). | ...<br>SOCIAL HISTORY:<br>«Residence»(LivingStatus, StatusTime-current): [LOCATION] «with husband and kids »(TypeLiving-with_family)<br>«Job»(Employment): «no longer works »(StatusEmploy-unemployed) |

Table 4: GPT-4 responses before and after post-processing.

## A.4 Heuristics for Trigger-argument Connections in the GPT-inline Outputs

We use the sample example in Table. 4 to demonstrate our heuristics for finding the trigger-arguments from GPT-inline outputs and output to BRAT standoff format. For example, after post-processing, the $T$ and $A$ arguments can be directly extracted:

> T1 LivingStatus 88 97 Residence
> T2 TypeLiving 110 131 with husband and kids
> A1 TypeLivingVal T2 with_family
> T3 StatusTime 88 97 Residence
> A2 StatusTimeVal T3 current
> T4 Employment 132 135 Job
> T5 StatusEmploy 137 152 no longer works
> A6 StatusEmployVal T5 unemployed
> ...

The above examples contain two trigger spans: T1 and T4. For the rest argument spans, we want to link each of them to its closest trigger span, constrained by the annotation guideline. For example, the distance between the argument T2 and the trigger T1 is 132 (T4 start) - 131 (T2 end) = 1 (character index), and the distance between T2 and the trigger T4 is 131 (T2 start) - 97 (T1 end) = 34 (character index). T1 is closer to the trigger T4. However, because the *TypeLiving* argument can only be attached to the *LivingStatus* trigger, T1 is attached to its closet *LivingStatus* trigger T4. Note that it is possible that a trigger does not contain any arguments or an argument is not attached to any trigger in GPT-inline outputs. Arguments in the above example can be summarized into BRAT events as:

> E1 LivingStatus:T1 Type:T2 Status:T3
> E2 Employment:T4 Status:T5
> ...