SMARTMiner: Extracting and Evaluating SMART Goals from Low-Resource Health Coaching Notes

Anonymous ACL submission

Abstract

We present SMARTMiner, a framework for extracting and evaluating SMART (Specific, Measurable, Attainable, Relevant, Time-bound) goals from unstructured health coaching (HC) notes. Developed in response to challenges observed during a clinical trial, the SMARTMiner achieves two tasks: (i) extracting behavior-change goal spans, and (ii) categorizing their SMARTness. We introduce SMARTSpan, the first publicly available dataset of 173 HC notes annotated with 266 goals and SMART attributes. SMARTMiner incorporates an extractive goal retriever with a component-wise SMART classifier. Experiment results show that extractive models significantly outperformed their generative counterparts in low-resource settings, and that two-stage fine-tuning substantially boosted performance. The classifier achieved up to 0.91 SMART F1 score, while the full SMARTMiner maintained high end-to-end accuracy. This work bridges healthcare, behavioral science, and natural language processing to support health coaches and clients with structured goal tracking—paving way for automated weekly goal reviews between human-led HC sessions. Code and the dataset will be released upon acceptance.

1 Introduction

Health coaching (HC) is a person-centered intervention designed to facilitate sustainable change in healthy behavior and support self-management of chronic diseases. Recent systematic reviews demonstrated that HC can significantly enhance physical activity, reduce pain, and improve psychological outcomes such as self-efficacy and quality of life among individuals with chronic conditions (Kastner et al., 2018; Yang et al., 2020a; Weiss et al.). A cornerstone of HC is the co-creation of actionable short-term goals that drive long-term behavior change, which in turn forms the foundation of its effectiveness as a behavioral intervention (Wallace et al., 2018).

Overall, the research suggests that specific and actionable goals tend to improve health behavior outcomes more than unclear or generic behavioral goals (Wallace et al., 2018; Bahrami et al., 2022), by providing focus and measurable targets. SMART (Specific, Measurable, Attainable, Relevant, and Time-bound) goals (Figure 1) often yield better short-term results and can help sustain behavior change (Doran, 1981; White et al., 2020), as seen in improved exercise levels, weight loss, and self-management behaviors in various studies (Olsen and Nesbitt, 2010; Wolever et al., 2010).

However, setting and evaluating SMART goals are laborious and complex (Bowman et al., 2015). Some goals may not be fully SMART, lacking one or more components. This may stem from patient's varying levels of readiness to engage in health-related behaviors (Prochaska and Velicer, 1997). Additionally, patients' recall of health behavior advice has been reported to be poor (Flocke and Stange, 2004), highlighting the need for improved goal documentation and patients support through the reinforcement of their goals between HC sessions, which usually span biweekly or monthly.

In response, this paper presents the development of a SMARTMiner framework that automatically extracts multiple (SMART) goals set during HC sessions from unstructured session notes. Since these goals are embedded within free-text narratives and cannot be audited at scale, we address two key challenges: **(i)** ***Goal extraction***: identifying multiple behavior-change goal spans within unstructured HC notes; and **(ii)** ***SMARTness diagnosis***: determining which SMART attributes each extracted goal satisfies and where it falls short.

Our contributions are three-fold:

1. **SMARTSpan corpus** – the first publicly available dataset of 173 HC notes with 266 unique goal spans, exhaustively annotated for both goal boundaries and SMART attributes.
2. **SMARTMiner framework** – a span extrac-

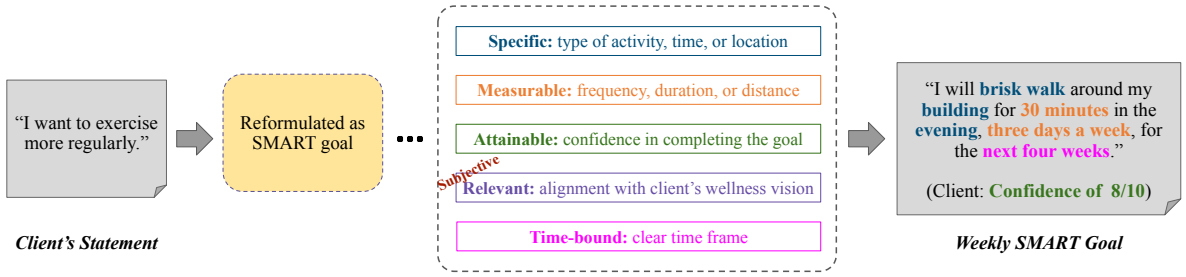

Figure 1: Reformulation of generic behavior goal into a SMART goal.

tor that locates every potential goal and an attribute-level classifier that flags missing SMART components, yielding actionable, interpretable feedback.

3. **Comprehensive evaluation and analysis** – five-fold in-domain and cross-domain experiments with diverse extractive and generative baselines reveal how low-resource, domain-specific data degrade state-of-the-art large language models (LLMs); a qualitative error taxonomy (hallucination, null extraction, attribute misclassification) pinpoints safety-critical failure modes for clinical deployment.

By combining healthcare, behavioral science, and natural language processing (NLP), the proposed SMARTMiner framework enables low-touch (SMART) goal tracking: it helps health coaches refine goals as needed and allows clients to review their (SMART) goals between HC sessions.

2 Related work

2.1 Clinical Notes

Information extraction (IE) from unstructured clinical notes has evolved from rule-based heuristics to transformer-based models capable of capturing a wide range of clinically relevant signals (Pai et al., 2024). In particular, ClinicalBERT detects medication gaps (Sarraju et al., 2022; Gobbel et al., 2022), while other models extract lifestyle and social factors (Zhou et al., 2019; Romanowski et al., 2023) or patient goals (Gupta et al., 2021). Beyond tagging, span-based models handle action items (Mullenbach et al., 2021), and temporal models align events chronologically (Miller et al., 2023). Prompt-driven LLMs such as GPT-4 now match or surpass supervised baselines with minimal data (Ramachandran et al., 2023; Agrawal et al., 2022), signaling a shift toward adaptable, low-resource IE solutions.

2.2 HC Conversations

Extracting SMART goals from HC conversations presents challenges at the intersection of clinical NLP and behavioral health (Chen and Hirschberg, 2024). Early approaches utilized rule-based systems and sequence labeling to identify SMART components within HC dialogues (Gupta et al., 2019, 2020b). Subsequent methods incorporated dialogue act modeling and transformer-based architectures to enhance goal extraction accuracy (Mullenbach et al., 2021; Gupta et al., 2020a,b). Recent studies proposed modularized and neuro-symbolic approaches to enhance goal summarization in low-resource settings, where labeled data is sparse and HC dialogues vary in format and structure (Zhou et al., 2022, 2024). However, the vagueness of conversational language calls for models that identify discontiguous spans and align to goal criteria, motivating span-based frameworks that are grounded in real-world HC interventions.

2.3 Multi-Span Reading Comprehension

Extracting multiple behavior-change goal spans from HC notes can be formulated as a Multi-span Reading Comprehension (MSRS) task. The existing MSRC methods fall into three categories:

- *Extractive* methods select answer spans from the input without generating new tokens, and are further divided into *token-based* and *span-based*. Token-based models predict answers through token-level outputs, where each token individually influences span selection (Hu et al., 2019; Yang et al., 2020b; Segal et al., 2020; Li et al., 2022; Luo et al., 2024). In contrast, span-based models explicitly score or classify candidate spans as wholes, considering span-level representations (Huang et al., 2023a; Zhang et al., 2024, 2023b).

- *Generative* methods, on the other hand, produce answers by generating tokens, often ex-

tending pre-trained generative models with fine-tuning strategies (Ai et al., 2024) or prompt engineering to adapt to the multi-span scenario (Mallick et al., 2023; Huang et al., 2023b; Zhang et al., 2023a).

- *Hybrid* methods leverage the strengths of both paradigms, either through unified frameworks (Lin et al., 2024) or via data augmentation techniques (Lee et al., 2023).

Most datasets used to evaluate models for MSRC focus on questions requiring the extraction of multiple discontiguous answer spans from text. Prominent benchmarks include *MultiSpanQA* (Li et al., 2022), which contains over 6,500 multi-span questions initially and around 19,000 in the expanded set; *QUOREF*, which comprises more than 24,000 questions requiring coreference resolution (Dasigi et al., 2019); and *DROP*, which includes around 96,000 questions involving arithmetic or reasoning over multiple spans (Dua et al., 2019). More specialized benchmark on the healthcare domain, *MASH-QA*, consists of approximately 35,000 question–answer pairs with long, multi-sentence answers (Zhu et al., 2020).

While existing MSRC models perform well on large, clean, and publicly available datasets (e.g., Wikipedia-based or curated medical content), such datasets are expensive and time-consuming to curate in clinical practice, including HC. As a result, span-centric models remain underexplored in low-resource, domain-specific settings. To address this gap, we introduce *SMARTSpan*, a curated dataset of HC notes with annotated behavioral goals, designed to support span-based goal extraction in practical, small-scale scenarios.

3 SMARTSpan Dataset

SMARTSpan dataset comprises 173 annotated HC notes from a randomized controlled trial (RCT) and is designed to evaluate multi-span extraction in low-resources, domain-specific scenarios.

3.1 Data Collection

The SMARTSpan dataset originates from an RCT aimed at prevention of cardiovascular disease through a multicomponent digital behavioral intervention, focusing on improving patient's adherence to statin therapy and promoting healthy behavioral change to reduce low-density lipoprotein (LDL) cholesterol levels.

One key component is human-led HC, where intervention participants receive six monthly coaching sessions via a mobile app. Clients are encouraged to set weekly SMART goals during each HC session, which are reviewed in the next session. After each session, health coaches document key observations and any SMART goals set with clients as unstructured free-text notes, without any given standardized template or guidance, on a web-based platform (details in Appendix A).

At the time of dataset creation, a mutltiracial cohort of approximately 130 patients with hyperlipidaemia had been enrolled in the ongoing RCT, with over 60 in the intervention arm receiving support from three certified health coaches. More than 180 HC sessions were conducted. Using a custom SQL query on the platform's backend, we retrieved 173 HC notes as the SMARTSpan dataset.

Anonymization. Protecting client privacy is essential when handling real-world HC notes with sensitive information. We adhered to an *iterative anonymization process* that combined human oversight with automated support by `ChatGPT-4o`. Personal identifiers were removed or replaced with "client", gendered language was neutralized, references to family members, locations, and dates were anonymized or deleted. Medical metrics (e.g., cholesterol, body weight) and personal attributes (e.g., age, occupation) were modified. Multiple review cycles ensured thorough anonymization while preserving data integrity and utility.

3.2 Data Annotation and Exploration

After anonymization, the dataset creation process was conducted in two phases.

Goal Annotation. In the first phase, two annotators manually reviewed 173 HC notes and identified goal statements within each note, marking any text that reflected specific behavioral objectives discussed or set during sessions. As shown in Table 1, the distribution of goals per note ranges from 0 to 5, depending on the depth and focus of the HC session. 27% of the HC notes contain no goals, and another 27% include only one. This distribution contrasts sharply with the MultiSpanQA dataset, where most questions have 2-3 answer spans (58% with 2, 22% with 3). SMARTSpan exhibits considerable sparsity in goal annotations, highlighting a major challenge in adapting existing MSRC techniques to real-world HC datasets, where relevant information is often infrequent and diverse.

# of answer spans or goals annotated	0	1	2	3	4–5	6–8	9–12	13–21
#Spans from MultiSpanQA	–	–	3,791	1,414	915	337	71	8
#Goals from SMARTSpan	46	46	37	32	12	–	–	–

Table 1: Comparison of the number of answer spans per question in MultiSpanQA (Li et al., 2022) and the number of goals per HC note in SMARTSpan dataset.

goals per HC note	0	1	2	3	4	5	$\sum$ Goals
Split_1	5	9	3	7	0	1	41
Split_2	4	7	4	8	2	0	47
Split_3	4	7	8	2	4	0	45
Split_4	7	6	5	4	2	1	41
Split_5	12	5	5	3	0	0	24

Table 2: Distribution of the number of goals per HC note and the total number of goals in the SMARTSpan test sets across five data splits.

SMARTness Annotation. In the second phase, each extracted goal was annotated for three core SMART components: Specific (S), Measurable (M), and Attainable (A). We excluded Relevant (R) due to its subjectivity and assumed Time-bound (T) was implicitly satisfied, as goals were intended to be achieved before the next HC session. Each of the three assessed components was annotated as a binary label (0 or 1). The final multiclass label was derived deterministically: *SMART* if all three were true, *Partially SMART* if two were true, and *Not SMART* if one or none were true.

To create a labeled dataset for supervised classification, two annotators with prior HC training independently rated the extracted goals. Disagreements were resolved through discussion. Inter-annotator agreement (IAA) was assessed using Cohen's kappa coefficient (Cohen, 1960) for each SMART components across 266 behavior goals. IAA was moderate for S ($\kappa = 0.574$, $z = 9.36$, $p < 0.001$), substantial for M ($\kappa = 0.812$, $z = 13.3$, $p < 0.001$), and near perfect for A ($\kappa = 1.00$, $z = 16.3$, $p < 0.001$). These results indicate statistically significant IAA beyond chance across all three components, with the highest consistency observed in component A. Additional details on annotation process are provided in Appendix B.

Cross-Validation Setup. Given the limited size of SMARTSpan (173 annotated HC notes with a total of 266 goals), we adopted a five-fold cross-validation for robust model evaluation, each follows a 70%/15%/15% (train/val/test) split. Each test and validation sets in splits contain 25 HC notes, with the remaining 123 HC notes used for training. As shown in Table 2, the number of goals per test split varies from 24 to 47 across splits, reflecting differences in goal density and highlighting the importance of evaluating model robustness.

4 Methods

We propose a SMARTMiner framework for multi-span behavioral SMART goal extraction from unstructured HC notes, as demonstrated in Figure 2. The *goal extraction* module formulates goal identification as a span-based question answering task, enabling the extraction of multiple goal mentions from free-text HC notes. The *SMARTness classification* module subsequently evaluates these extracted goals along three key dimensions to determine their alignment with the SMART criteria.

4.1 Goal Extraction Module

Several extractive models were implemented within the goal extraction module by formulating the extraction task as an MSQA problem. Each HC note is treated as the context $C = \{c_1, c_2, \ldots, c_n\}$, and paired with a fixed-style question Q: *"What are the goals mentioned in the text?"* Although the original dataset includes annotated goal spans, it does not contain varying questions. To enable span-based supervision, we recast the dataset into a QA format using this fixed question. The model is then trained to extract all non-overlapping spans $P = \{p_1, p_2, \ldots, p_t\}$ that correspond to goals

$$P = M(C, Q). \tag{1}$$

Here, M denotes the fine-tuned extractive model. For generative models, we fine-tuned them using a prompt template with *instruction*, *input* and *output* (see Appendix C for the exact format).

4.2 SMARTness Classification Module

Classification Objective. Given an extracted goal span p_i, the classifier independently predicts binary values for each component

$$v_i^{(S)},\ v_i^{(M)},\ v_i^{(A)} = \text{Classifier}(p_i), \tag{2}$$

where $v_i^{(d)} \in \{0, 1\}$ for each dimension $d \in \{\text{S}, \text{M}, \text{A}\}$. These binary predictions are obtained via sigmoid-activated heads and trained using binary cross-entropy loss for each dimension. The final multiclass label is assigned post-hoc via rule-based aggregation over the binary predictions for each component. Specifically, we define

$$y_i = f\left(v_i^{(S)} + v_i^{(M)} + v_i^{(A)}\right), \tag{3}$$

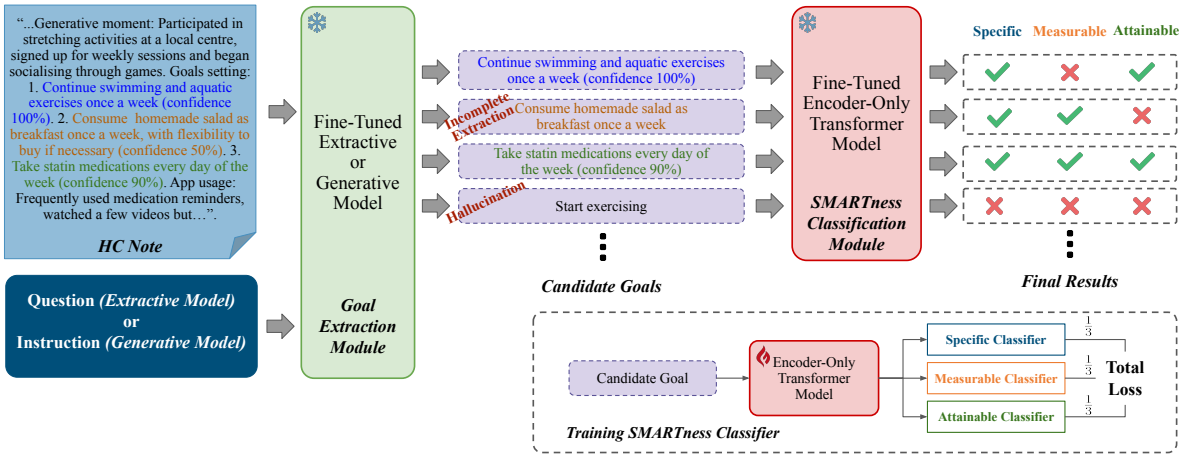

Figure 2: The overall architecture of our proposed SMARTMiner framework.

where $v_i^{(d)} \in \{0, 1\}$ denotes the binary prediction for SMART dimension $d \in \{\mathrm{S}, \mathrm{M}, \mathrm{A}\}$ and $f(\cdot)$ is a deterministic mapping from component count to structured label. A sum of 3 indicates a *SMART* goal; 2, *Partially SMART*; and 0 or 1, *Not SMART*.

Model Architecture. The classifier is implemented as a transformer-based model. We employed a pre-trained encoder to obtain contextualized representations of the input goal. Specifically, the embedding of the `[CLS]` token is extracted and passed through a shared processing stack consisting of dropout, a linear projection layer, and ReLU activation to generate a latent representation. This representation is then fed into three independent binary classification heads—implemented as linear layers with sigmoid activation—to estimate the probabilities $P^{(S)}$, $P^{(M)}$, and $P^{(A)}$ that the goal is Specific, Measurable, and Attainable, respectively.

Training and Loss. We trained the classifier using binary cross-entropy (BCE) loss independently for each SMART component. The overall training objective is defined as

$$\mathcal{L} = \frac{1}{3} \sum_{d} \mathrm{BCE}(v^{(d)}, y^{(d)}), \tag{4}$$

where $v^{(d)}$ is the predicted probability and $y^{(d)}$ the ground truth label for SMART dimension $d \in \{\mathrm{S}, \mathrm{M}, \mathrm{A}\}$. This setup enables the model to learn each structural dimension independently while supporting interpretable component-level diagnostics.

5 Experiment Setting

Datasets. To evaluate the effectiveness of our proposed SMARTMiner framework for multi-span behavioral SMART goal extraction in a low-resource setting, we used the SMARTSpan dataset described in Section 3. As no other existing corpus annotates SMART-goal spans, we also evaluate on MultiSpanQA (Li et al., 2022), a widely used MSRC benchmark whose "find-all-spans" setup mirrors our extraction task. This positions our model against established baselines while making clear the domain shift from open-domain question answering to the HC notes.

Evaluation Metrics. For goal extraction, we adopt both Exact Match (EM) and Partial Match (PM) Precision, Recall, and F1 as the primary evaluation metrics, following Li et al. (2022). For SMARTness classification, we report accuracy, macro-averaged F1, and the class-wise F1 score for the SMART label. These are computed over the three predicted classes: SMART, Partially SMART, and Not SMART.

Goal Extraction Module. We implemented and evaluated both extractive and generative approaches as the goal extraction modules. Since prior research suggests that span-centric methods outperform token-centric ones in multi-span settings Huang et al. (2023a); Zhang et al. (2024, 2023b), we selected two representative extractive strategies: *SpanQualifier* (Huang et al., 2023a), which scores candidate spans, and the *Contrastive Span Selector (CSS)* (Zhang et al., 2023b), which ranks spans using contrastive learning with positive and negative contextual cues. All models were fine-tuned using a single NVIDIA A40 GPU, with the exception of CSS on MultiSpanQA, which was fine-tuned using 8 NVIDIA A40 GPUs.

For *SpanQualifier*, we adopted the same configu-

ration as reported in (Huang et al., 2023a), with the exception of setting the same random seed to ensure consistency across all experiments. Given initially low performance when fine-tuning directly on the in-domain SMARTSpan dataset, we also experimented with a two-stage fine-tuning strategy: first pretraining on the MultiSpanQA dataset, followed by continued fine-tuning on SMARTSpan. We evaluated this approach using two pretrained language models: `bert-base-uncased` (Devlin et al., 2019) and `deberta-v3-base` (He et al., 2020).

For *CSS*, we adopted the same configuration as reported in (Zhang et al., 2023b), except for setting the random seed to 30 for consistency with other experiments and increasing the number of training epochs to 20. We evaluated this approach using two pretrained language models: `bert-base-uncased` and `roberta-base` (Liu et al., 2019).

To evaluate the performance of generative language models in low-resource, domain-specific goal extraction tasks, we fine-tuned decoder-only and encoder-decoder architectures (see Appendix D for the full list) using LoRA-based parameter-efficient adaptation (Hu et al., 2021). Fine-tuning was performed using the `Unsloth` library (Unsloth, 2024), which is optimized for reduced memory usage and faster training. We applied LoRA with a rank of 8 and an alpha scaling factor of 32, enabling efficient adaptation of LLMs while maintaining performance. Each pretrained model was fine-tuned on a single GPU for 20 epochs, using a batch size of 4, a learning rate of 10^{-4}, and a weight decay of 0.01. We employed AdamW with 8-bit optimization, linear learning rate scheduling, mixed-precision training, and gradient checkpointing. Early stopping was used with a patience of 5 valid steps to prevent overfitting.

We also adapted the QASE framework (Ai et al., 2024) to enable span-level supervision in generative models. While we followed the original hyperparameter settings proposed by the authors, we introduced two key modifications. First, we set the batch size to 4 across all experiments to fit within our GPU memory constraints. Second, given the smaller size of our target dataset, we increased the number of training epochs to 20. For the larger MultiSpanQA dataset, however, we retained the original configuration of training for 3 epochs.

Finally, we evaluated a zero-shot, schema-based prompting approach using `GPT-4.1` (OpenAI, 2023) as a non-fine-tuned baseline. The model was guided by structured extraction instructions, using the same instruction template as that for the fine-tuned generative models, with a single modification: the final sentence was changed from "*Format your response as a numbered list.*" to "*Always respond in JSON format.*", which ensures compatibility with schema-based function calling. The model was deployed via OpenAI's function-calling API to extract weekly SMART goals from unstructured HC notes. SMART goal extraction was treated as a semantic parsing task, with outputs constrained to a predefined JSON schema to enforce structural compliance (see Appendix E).

SMARTness Classification Module. We fine-tuned three transformer-based models: `deberta-v3-base`, `deberta-v3-large`, and `roberta-large`. Each model was trained using a batch size of 4, a maximum sequence length of 64, and a learning rate of 2×10^{-5}. Optimization was performed using AdamW with a weight decay of 0.01. Training proceeded for up to 20 epochs with early stopping based on validation loss, using a patience threshold of 5. Random seed was fixed to ensure stability and reproducibility.

6 Results and Discussion

Table 3 compares the performance of different models on the SMARTSpan and MultiSpanQA datasets for goal extraction, while Table 4 summarizes the results of three encoder-based models fine-tuned on SMARTSpan for SMARTness classification. In both tables, results on SMARTSpan are reported as mean and standard deviation, as all models were evaluated across five test splits as described in Section 3.2. Full split-wise results for all models are provided in Appendix F and Appendix G.

6.1 Performance of Extractive Models on Goal Extraction Task

Extractive models generally achieve the highest performance on the goal extraction task across both datasets (Table 3). The best-performing model on SMARTSpan is the `DeBERTa-v3-base` fine-tuned sequentially on MultiSpanQA and SMARTSpan using the SpanQualifier framework, which establishes a strong baseline for span-based goal extraction. However, extractive models fail to generalize when trained solely on SMARTSpan. Specifically, when `DeBERTa-v3-base` is fine-tuned only on SMARTSpan, its performance drops significantly, resulting in the worst performance among all evaluated models.

Model	#Params	SMARTSpan						MultiSpanQA					
		EM (mean ± sd)			PM (mean ± sd)			EM			PM		
		P↑	R↑	F↑	P↑	R↑	F↑	P↑	R↑	F↑	P↑	R↑	F↑
Extractive models (SpanQualifier)													
DeBERTa-v3-base$_{\text{SMARTSpan}}$		0.28 (0.17)	20.94 (10.36)	0.56 (0.35)	22.20 (3.12)	75.79 (4.49)	34.26 (3.39)	–	–	–	–	–	–
DeBERTa-v3-base$_{\text{MultiSpanQA_SMARTSpan}}$	180M	**85.24** (4.89)	84.46 (5.00)	**84.83** (4.83)	**93.69** (4.48)	90.72 (3.70)	**92.14** (3.57)	–	–	–	–	–	–
DeBERTa-v3-base$_{\text{MultiSpanQA}}$		12.54 (5.00)	13.16 (6.27)	12.81 (5.60)	39.55 (5.85)	27.44 (7.86)	32.11 (7.09)	**76.56**	73.31	**74.90**	88.49	83.37	85.86
bert-base-uncased$_{\text{SMARTSpan}}$		0.34 (0.13)	24.43 (6.86)	0.68 (0.24)	23.35 (1.96)	79.42 (5.75)	36.07 (2.86)	–	–	–	–	–	–
bert-base-uncased$_{\text{MultiSpanQA_SMARTSpan}}$	110M	78.96 (1.41)	74.78 (2.04)	76.80 (1.56)	89.37 (3.31)	84.24 (2.02)	86.72 (2.47)	–	–	–	–	–	–
bert-base-uncased$_{\text{MultiSpanQA}}$		9.47 (4.98)	10.58 (6.02)	9.92 (5.32)	35.73 (5.92)	22.07 (5.42)	26.79 (4.48)	66.99	68.81	67.89	80.07	78.17	79.11
Extractive models (CSS)													
roberta-base	125M	65.80 (10.22)	**88.12** (3.28)	74.84 (6.72)	79.24 (8.56)	**92.80** (2.97)	85.14 (4.58)	74.93	69.91	72.33	85.95	77.51	81.51
bert-base-uncased	110M	68.00 (7.65)	84.85 (3.99)	75.23 (5.45)	82.46 (6.86)	89.18 (4.50)	85.56 (5.04)	69.93	61.22	65.29	81.82	70.26	75.60
Generative models (LORA fine-tuning)													
phi-4	14B	32.58 (15.79)	24.75 (18.18)	27.69 (17.40)	42.21 (21.90)	32.16 (23.81)	35.96 (23.17)	74.28	73.00	73.63	88.96	84.71	86.78
Mistral-Nemo-Instruct-2407	13B	35.59 (10.95)	26.84 (15.10)	30.14 (13.75)	52.49 (19.94)	40.35 (23.82)	45.01 (22.68)	29.60	**75.56**	42.53	47.39	**91.17**	62.36
gemma-2-9b-it	9B	36.65 (14.08)	37.71 (12.21)	36.40 (12.99)	62.98 (13.83)	65.99 (17.72)	62.89 (12.78)	68.90	75.46	72.03	83.68	87.57	85.58
Meta-Llama-3.1-8B	8B	28.67 (11.19)	18.09 (9.22)	22.08 (10.26)	33.98 (14.30)	21.74 (11.87)	26.39 (13.14)	55.86	57.14	56.49	73.26	80.09	76.52
mistral-7b-instruct-v0.3	7B	46.30 (8.10)	49.61 (14.97)	47.52 (11.13)	74.85 (8.24)	77.49 (16.99)	75.63 (12.03)	72.84	71.59	72.21	87.74	84.09	85.87
Llama-3.2-3B	3B	24.52 (5.88)	33.61 (10.71)	27.64 (6.35)	43.33 (10.11)	51.87 (9.36)	46.16 (8.20)	65.82	71.95	68.75	81.52	85.50	83.46
flan-t5-large	750M	44.38 (10.44)	37.80 (6.09)	40.57 (7.45)	66.54 (10.88)	78.18 (4.20)	71.54 (7.67)	74.03	71.90	72.95	88.01	85.40	86.68
bart-large	406M	34.80 (10.91)	23.65 (8.32)	28.14 (9.46)	69.79 (12.15)	55.71 (6.34)	61.70 (8.36)	48.92	47.56	48.23	65.30	59.93	62.50
Generative models (QASE)													
flan-t5-large	750M	35.84 (4.92)	28.89 (3.02)	31.89 (3.44)	70.35 (8.00)	50.01 (1.57)	58.34 (3.88)	75.59	70.17	72.78	**91.48**	83.12	**87.10**
Generative models (LLM Schema)													
GPT-4.1	–	39.89 (8.52)	34.92 (9.28)	37.14 (8.84)	84.57 (6.31)	67.42 (6.02)	74.88 (5.40)	–	–	–	–	–	–

Table 3: Performance evaluation on SMARTSpan and MultiSpanQA datasets for all Goal Extraction models, sorted by parameter size (descending). All metrics are higher-the-better (↑) and numbers in parentheses correspond to standard deviation. The best result per column is in **bold**, second-best is underlined.

Model	#Params	Accuracy (↑)	Macro F1 (↑)	SMART F1 (↑)
deberta-v3-large	435M	**0.86 ± 0.02**	**0.85 ± 0.03**	**0.91 ± 0.03**
roberta-large	355M	0.70 ± 0.26	0.67 ± 0.30	0.70 ± 0.40
deberta-v3-base	180M	0.83 ± 0.05	0.81 ± 0.06	0.90 ± 0.03

Table 4: Evaluation on SMARTSpan dataset for all SMARTness Classification models, sorted by parameter size (descending). All metrics are higher-the-better (↑). Best results per column are in **bold**.

This significant performance reduction highlights the sensitivity of extractive models to limited training data. The SMARTSpan training split comprises only 123 examples, which appears to be insufficient for the model to learn effective span representations without prior exposure to larger datasets like MultiSpanQA (see Appendix H). Despite this, the two-stage fine-tuning process mitigates the performance gap. As shown in Appendix J, models first exposed to MultiSpanQA can successfully identify goal-relevant regions even in loosely structured HC notes, demonstrating the importance of pretraining on richly supervised multi-span data before adapting to low-resource, domain-specific datasets such as SMARTSpan.

6.2 Performance of Generative Models on Goal Extraction Task

Generative LLMs consistently underperform span-extractive baselines on SMARTSpan. For instance, `mistral-7b-instruct-v0.3` achieves only 47.52 EM and 75.63 PM F1 score, whereas a *20× smaller* CSS extractor achieves 75.23 EM and 85.56 PM F1 score. Their larger split-to-split standard deviations provide further evidence of their instability under limited training data. In contrast, the same generative models match or surpass extractive systems on MultiSpanQA (up to 87.10 PM F1 score), highlighting a strong sensitivity to domain shift from open-domain MSRC to low-resource, health-coaching notes. Until tighter span-faithfulness constraints emerge, extractive or hybrid pipelines remain the safer choice for clinical goal extraction.

Manual inspection further pinpoints two recurrent failure modes in generative outputs: (i) hallucination – insertion of content absent from the note (e.g., random HTML tags, stray characters, and fabricated text; see Appendices J and K); (ii) null extraction – Span-grounding methods such as QASE (Ai et al., 2024) reduce these errors on MultiSpanQA, yet prove ineffective on SMARTSpan, leaving safety-critical, responsibility-sensitive failures largely unaddressed.

6.3 Analysis of the SMARTness Classifiers

As shown in Table 4, we evaluated three transformer-based models trained on SMARTSpan for SMARTness classification: `DeBERTa-v3-large`, `RoBERTa-large`, and

DeBERTa-v3-base. DeBERTa-v3-large achieved the highest scores on all evaluation metrics. Its accuracy reached 0.86, with a macro-average F1 of 0.85 and a SMART F1 score of 0.91. DeBERTa-v3-base is a smaller model but performed nearly as well in all categories. Its SMART F1 was only modestly lower, which suggests that it may serve as a practical alternative when computational resources are limited.

To gain a better understanding of the remaining mistakes, we reviewed the predictions and grouped the errors into three common types. The first is boundary confusion, where the model struggles to distinguish between categories such as SMART and Partially SMART. This often occurs when a goal is nearly completed but lacks a minor element, such as a specific time frame. The second type is overclassification, where incomplete goals are incorrectly labeled as SMART. This usually occurs when vague wording or loosely defined measures are interpreted as sufficient. The third type is underclassification, which refers to clearly defined SMART goals being labeled as less specific because key attributes are implied rather than being stated directly. These observations suggest that classification performance could be improved by including more training examples that reflect subtle variations in goal phrasing. It may also help to better define how each SMART attribute should be recognized during training.

6.4 Performance Evaluation of the SMARTMiner Framework

Finally, we evaluated our SMARTMiner framework by combining the best-performing goal extraction model (DeBERTa-v3-base) with a SMARTness classification model based on DeBERTa-v3-large. Inference was conducted across all five SMARTSpan test splits, yielding a total of 198 golden goals, of which 76% were unique. Table 5 presents the SMARTness confusion matrix, reflecting the classification applied to goals extracted by the goal extraction module, rather than gold-standard annotations. The framework demonstrates strong overall performance, particularly in correctly identifying SMART goals.

Most misclassifications occurred between adjacent categories—for example, Partially SMART goals labeled as SMART. To further examine system behavior, we conducted a focused analysis of the most extreme misclassifications—cases where SMART goals were incorrectly predicted as Not SMART (underprediction), and vice versa (overprediction). The underprediction stemmed from incomplete span extraction, omitting key information. Three overpredictions were caused by hallucinated goals, and one involved a correctly extracted goal with overestimated SMARTness. These findings underscore how extraction errors can cascade into misclassification, highlighting the need for precision across both modules.

label \ pred	Not SMART	Partially SMART	SMART
Not SMART	**47 (73.44%)**	13 (20.31%)	4 (6.25%)
Partially SMART	6 (10.71%)	**39 (69.64%)**	11 (19.64%)
SMART	1 (1.09%)	9 (9.78%)	**82 (89.13%)**

Table 5: Performance evaluation of the SMARTMiner framework across all five SMARTSpan test splits. Correct predictions are shown in **bold**.

7 Conclusions and Future Work

This paper presents a SMARTMiner framework to support health coaches in identifying goals that may require reformulation, while also surfacing all goals to clients via their mobile app. Our evaluation revealed that the performance gap between extractive span-based methods such as SpanQualifier and fine-tuned generative models is more pronounced in real-world, low-resource settings such as SMARTSpan than on larger datasets such as MultiSpanQA. Nevertheless, we demonstrated that, by carefully selecting base models and adopting a two-stage fine-tuning approach, the proposed framework can achieve good performance in identifying and categorizing various types of goals.

Building on this foundation, we envision extending SMARTMiner to a fully automated system that conducts weekly goal reviews between human-led HC sessions. This system would operate independently, triggering structured check-ins to assess progress, reinforce accountability, and provide timely feedback to health coaches regarding their clients' progress. By integrating this functionality, the system aims to sustain engagement and support behavior change more consistently over time, even in the absence of direct human interaction. Expanding this framework further, it could potentially be applied to other health and social care contexts such as documenting and tracking individual care plans, health and social prescribing programmes.

Limitations

As stated in Section 3, for anonymization of our data, we employed ChatGPT only for *minimal, word-level rewrite*; every rewrite suggestion was reviewed and, where necessary, corrected by human annotators to ensure that no personally identifiable information remained and that the note structure, wording, and formatting were preserved exactly, avoiding hallucinated edits or template drift. This intensive human-in-the-loop pipeline is feasible for the present corpus of 173 HC notes, but does not scale linearly. For larger releases we will introduce an additional annotation layer that *grades* each LLM-generated rewrite for fidelity and formatting compliance before acceptance, allowing us to maintain real-world note realism while keeping human effort tractable.

Given the nature of the collected data (i.e., HC notes), we focused only on labeling the three core SMART components: Specific (S), Measurable (M), and Attainable (A). The Time-bound (T) element was implicitly defined by the structure of the intervention, where clients set weekly goals or targets to be completed before the next HC session. However, since this temporal aspect was rarely stated explicitly in the HC notes, it could not be annotated as a gold label or reliably captured by our classifier. Similarly, the Relevant (R) dimension was excluded from annotation, as health coaches served as the first filter—only documenting goals that were deemed relevant within the scope of the study. As a result, R was not labeled in the dataset and thus could not be assessed.

All HC notes in the SMARTSpan dataset are written in English and were collected over a one-year period from a specific population with elevated LDL levels, participating in an intervention to improve adherence to statin therapy. The dataset reflects the language, behaviors, and health system context of this particular group. Anyone using the dataset should be mindful of these contextual limitations when applying the models or generalizing findings to other populations or settings.

Ethical Considerations

The RCT from which the HC session notes were obtained received ethical approval from the Institutional Review Board (IRB). Written informed consent was obtained from all participants enrolled in the RCT. We will release the full details of IRB approval after the peer-review process.

The use of automated framework for SMART goal extraction and evaluation carries several potential risks. The models may overlook contextual nuances such as sarcasm, conditional statements, or culturally sensitive expressions, leading to distorted representations of the user's intent and potentially triggering inappropriate follow-ups. Over-reliance on model outputs without human verification could compromise care quality, particularly if health coaches treat the model's evaluations as authoritative. Additionally, the framework may inherit and amplify biases from the training data, favoring certain goal formats or under-representing the linguistic patterns of specific population groups. This underscores the essentiality of impartiality in clinical documentation to enhance the precision of the current framework and similar frameworks in the future.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lin Ai, Zheng Hui, Zizhou Liu, and Julia Hirschberg. 2024. Enhancing pre-trained generative language models with question attended span extraction on machine reading comprehension. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10046–10063.

Zeynab Bahrami, Atena Heidari, and Jacquelyn Cranney. 2022. Applying smart goal intervention leads to greater goal attainment, need satisfaction and positive affect. *International Journal of Mental Health Promotion*, 24(6).

Julia Bowman, Lise Mogensen, Elisabeth Marsland, and Natasha Lannin. 2015. The development, content validity and inter-rater reliability of the smart-goal evaluation method: A standardised method for evaluating clinical goals. *Australian occupational therapy journal*, 62(6):420–427.

Yu-Wen Chen and Julia Hirschberg. 2024. Exploring robustness in doctor-patient conversation summarization: An analysis of out-of-domain SOAP notes. In *Proceedings of the 6th Clinical Natural Language*

Processing Workshop, pages 1–9, Mexico City, Mexico. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Pradeep Dasigi, Nelson F Liu, Ana Marasović, Noah A Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

George T Doran. 1981. There's a smart way to write managements's goals and objectives. *Management review*, 70(11).

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Susan A Flocke and Kurt C Stange. 2004. Direct observation and patient recall of health behavior advice. *Preventive medicine*, 38(3):343–349.

Glenn T Gobbel, Michael E Matheny, Ruth R Reeves, Julia M Akeroyd, Alexander Turchin, Christie M Ballantyne, Laura A Petersen, and Salim S Virani. 2022. Leveraging structured and unstructured electronic health record data to detect reasons for suboptimal statin therapy use in patients with atherosclerotic cardiovascular disease. *American Journal of Preventive Cardiology*, 9:100300.

Itika Gupta, Barbara Di Eugenio, Brian Ziebart, Aiswarya Baiju, Bing Liu, Ben Gerber, Lisa Sharp, Nadia Nabulsi, and Mary Smart. 2020a. Human-human health coaching via text messages: Corpus, annotation, and analysis. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 246–256, 1st virtual meeting. Association for Computational Linguistics.

Itika Gupta, Barbara Di Eugenio, Brian Ziebart, Bing Liu, Ben Gerber, and Lisa Sharp. 2019. Modeling health coaching dialogues for behavioral goal extraction. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1188–1190.

Itika Gupta, Barbara Di Eugenio, Brian Ziebart, Bing Liu, Ben Gerber, and Lisa Sharp. 2020b. Goal summarization for human-human health coaching dialogues. In *The Thirty-Third International FLAIRS Conference (FLAIRS-33)*.

Itika Gupta, Barbara Di Eugenio, Brian D. Ziebart, Bing Liu, Ben S. Gerber, and Lisa K. Sharp. 2021. Summarizing behavioral change goals from SMS exchanges to support health coaches. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–289, Singapore and Online. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. A multi-type multi-span network for reading comprehension that requires discrete reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1596–1606.

Zixian Huang, Jiaying Zhou, Chenxu Niu, and Gong Cheng. 2023a. Spans, not tokens: a span-centric model for multi-span reading comprehension. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 874–884.

Zixian Huang, Jiaying Zhou, Gengyang Xiao, and Gong Cheng. 2023b. Enhancing in-context learning with answer feedback for multi-span question answering. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 744–756. Springer.

Monika Kastner, Roberta Cardoso, Yonda Lai, Victoria Treister, Jemila S Hamid, Leigh Hayden, Geoff Wong, Noah M Ivers, Barbara Liu, Sharon Marr, and 1 others. 2018. Effectiveness of interventions for managing multiple high-burden chronic diseases in older adults: a systematic review and meta-analysis. *Cmaj*, 190(34):E1004–E1012.

Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. 2023. Liquid: a framework for list question answering

dataset generation. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, pages 13014–13024.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022. Multispanqa: A dataset for multi-span question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260.

Jiayi Lin, Chenyang Zhang, Haibo Tong, Dongyu Zhang, Qingqing Hong, Bingxuan Hou, and Junli Wang. 2024. Correct after answer: Enhancing multi-span question answering with post-processing method. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2701–2717.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhiyi Luo, Yingying Zhang, and Shuyun Luo. 2024. A token-based transition-aware joint framework for multi-span question answering. *Information Processing & Management*, 61(3):103678.

Prabir Mallick, Tapas Nayak, and Indrajit Bhattacharya. 2023. Adapting pre-trained generative models for extractive question answering. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 128–137.

Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open models. Accessed: February 3, 2025.

Timothy Miller, Steven Bethard, Dmitriy Dligach, and Guergana Savova. 2023. End-to-end clinical temporal information extraction with multi-head attention. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 313–319, Toronto, Canada. Association for Computational Linguistics.

James Mullenbach, Yada Pruksachatkun, Sean Adler, Jennifer Seale, Jordan Swartz, Greg McKelvey, Hui Dai, Yi Yang, and David Sontag. 2021. CLIP: A dataset for extracting action items for physicians from hospital discharge notes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1365–1378, Online. Association for Computational Linguistics.

Jeanette M Olsen and Bonnie J Nesbitt. 2010. Health coaching to improve healthy lifestyle behaviors: an integrative review. *American journal of health promotion*, 25(1):e1–e12.

OpenAI. 2023. Gpt-4 technical report. https://openai.com/research/gpt-4. Accessed: 2025-05-08.

Liu Pai, Wenyang Gao, Wenjie Dong, Lin Ai, Ziwei Gong, Songfang Huang, Li Zongsheng, Ehsan Hoque, Julia Hirschberg, and Yue Zhang. 2024. A survey on open information extraction from rule-based model to large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9586–9608, Miami, Florida, USA. Association for Computational Linguistics.

James O Prochaska and Wayne F Velicer. 1997. The transtheoretical model of health behavior change. *American journal of health promotion*, 12(1):38–48.

Giridhar Kaushik Ramachandran, Yujuan Fu, Bin Han, Kevin Lybarger, Nic Dobbins, Ozlem Uzuner, and Meliha Yetisgen. 2023. Prompt-based extraction of social determinants of health using few-shot learning. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 385–393, Toronto, Canada. Association for Computational Linguistics.

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*, 30(8):1448–1455.

Ashish Sarraju, Jean Coquet, Alban Zammit, Antonia Chan, Summer Ngo, Tina Hernandez-Boussard, and Fatima Rodriguez. 2022. Using deep learning-based natural language processing to identify reasons for statin nonuse in patients with atherosclerotic cardiovascular disease. *Communications Medicine*, 2(1):88.

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2020. A simple and effective model for answering multi-span questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3074–3080.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Unsloth. 2024. Unsloth: Efficient fine-tuning of llms. Accessed: May. 8, 2025.

Anne M Wallace, Matthew T Bogard, and Susan M Zbikowski. 2018. Intrapersonal variation in goal setting and achievement in health coaching: cross-sectional retrospective analysis. *Journal of Medical Internet Research*, 20(1):e32.

Jocelyn M Weiss, Bala Munipalli, Miranda P Kaye, Katherine Smith, Eli Shur, Sebastian Harenberg, Rachel Garofalo, Arya B Mohabbat, Arden Robinson,

Stefan N Paul, and 1 others. Compendium of health and wellness coaching: 2023 addendum. *Journal of integrative and complementary medicine*.

Nicole D White, Vicki Bautista, Thomas Lenz, and Amy Cosimano. 2020. Using the smart-est goals in lifestyle medicine prescription. *American journal of lifestyle medicine*, 14(3):271–273.

RQ Wolever, M Dreusicke, J Fikkan, TV Hawkins, S Yeung, J Wakefield, L Duda, P Flowers, C Cook, and E Skinner. 2010. Integrative health coaching for patients with type 2 diabetes. *The Diabetes Educator*, 36(4):629–639.

Juan Yang, Brent A Bauer, Stephanie A Lindeen, Adam I Perlman, Abd Moain Abu Dabrh, Kasey R Boehmer, Manisha Salinas, and Susanne M Cutshall. 2020a. Current trends in health coaching for chronic conditions: A systematic review and meta-analysis of randomized controlled trials. *Medicine*, 99(30):e21080.

Junjie Yang, Zhuosheng Zhang, and Hai Zhao. 2020b. Multi-span style extraction for generative reading comprehension. *arXiv preprint arXiv:2009.07382*.

Chen Zhang, Jiuheng Lin, Xiao Liu, Yuxuan Lai, Yansong Feng, and Dongyan Zhao. 2023a. How many answers should i give? an empirical study of multi-answer reading comprehension. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5811–5827.

Penghui Zhang, Guanming Xiong, and Wen Zhao. 2023b. Css: Contrastive span selector for multi-span question answering. In *Pacific Rim International Conference on Artificial Intelligence*, pages 225–236. Springer.

Yingying Zhang, Zhiyi Luo, and Zuohua Ding. 2024. A simple and effective span interaction modeling method for enhancing multiple span question answering. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 188–200. Springer.

Xin Zhou, Yanshan Wang, Sunghwan Sohn, Terry M Therneau, Hongfang Liu, and David S Knopman. 2019. Automatic extraction and assessment of lifestyle exposures for alzheimer's disease using natural language processing. *International journal of medical informatics*, 130:103943.

Yue Zhou, Barbara Di Eugenio, Brian Ziebart, Lisa Sharp, Bing Liu, and Nikolaos Agadakos. 2024. Modeling low-resource health coaching dialogues via neuro-symbolic goal summarization and text-units-text generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11498–11509, Torino, Italia. ELRA and ICCL.

Yue Zhou, Barbara Di Eugenio, Brian Ziebart, Lisa Sharp, Bing Liu, Ben Gerber, Nikolaos Agadakos, and Shweta Yadav. 2022. Towards enhancing health coaching dialogue in low-resource settings. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 694–706, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K Reddy. 2020. Question answering with long multiple-span answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3840–3849.

A Example of HC Note

Figure 3 shows an example of a real-world HC note inputted into the HC web-based platform. The note includes SMART goals, lifestyle observations (e.g., travel), and structured follow-up on weekly SMART goal performance with patient-reported adherence and perceived success rates.

HC note

```
1. Swimming and aquarobics for 30 minutes once a week.
2. Homemade salad for breakfast once a week, buy salad as
alternative if busy.
3. Take medications every day of the week.

Client recently returned from a trip with a friend that
included food outings and leisure travel.

Goals review:
1. Aquatic exercises were done once a week due to work
commitments, perceived success 40-50%.
2. Had breakfast three times a week, usually bread and
coffee, perceived success 100%.
3. Took medications 6 out of 7 days, found pillbox and
app reminders helpful especially after late shifts,
perceived success 100%.

Generative moment: Participated in stretching activities
at a local centre, signed up for weekly sessions and began
socialising through games.

Goals setting:
1. Continue swimming and aquatic exercises once a week
(confidence 100%).
2. Consume  homemade salad as breakfast once a week,
with flexibility to buy if necessary (confidence 50%).
3. Take statin medications every day of the week
(confidence 90%).

App usage: Frequently used medication reminders, watched
a few videos but cited lack of time.

Other concerns: Currently adjusting antihypertensive
medication under physician's guidance due to low heart rate
and elevated BP readings. Client was instructed to monitor
and record BP and heart rate daily using the app diary and
dictation tool.
```

Extracted goals

The following goals were manually extracted from the above-mentioned HC note as golden labels

```
1. Continue swimming and aquatic exercises once a week
(confidence 100%)
2. Consume homemade salad as breakfast once a week, with
flexibility to buy if necessary (confidence 50%)
3. Take statin medications every day of the week
(confidence 90%)
```

SMARTness of extractive goals

Following our core SMARTness framework, the extracted goals are manually evaluated for Specificity, Measurability, and Attainability. A label of `[1,1,1]` indicates that the goal meets all three core criteria and it is labeled as SMART.

```
1. [1,0,1]
2. [1,1,1]
3. [1,1,1]
```

In the example above, the first goal is *Specific* (it mentions swimming and aquatic exercises) and *Attainable* (confidence of 100%), but not *Measurable*, as it does not specify the duration of swimming or aquatic activities. The second goal is *Specific* (consume homemade salad), *Measurable* (once a week), and *Attainable* (confidence assessed). Finally, the third goal is *Specific* (statin medications), *Measurable* (every day of the week), and *Attainable* (confidence of 90%). Some other examples of Partially SMART and Not SMART goals in the dataset are

```
1. Complete 5km walk two times a week from midweek onwards.
[1,1,0]
2. Exercise daily for at least 1 hour. [0,1,0]
3. Exercise daily for 30 minutes (confidence level 8/10).
[0,1,1]
4. Reduce portion sizes and adjust ingredients to
healthier options. [0,0,0]
```

Goal 1 is Partially SMART as it is *Specific* (5km walk) and *Measurable* (twice a week), but lacks an *Attainability* marker such as confidence or feasibility. Goal 2 is not SMART as it is *Measurable* (daily, 1 hour), but lacks *Specificity* (type of exercise) and does not include *Attainability*. Goal 3 is Partially SMART as it is *Measurable* (daily, 30 minutes) and *Attainable* (confidence of 8/10), but not *Specific* about the type of activity. Goal 4 is Not SMART as it is vaguely set and lacks all three SMART components—there is no clear target behavior, no quantifiable element, and no indication of feasibility.

B Annotation of SMARTSpan Dataset

The annotation was performed by two academic researchers who hold a doctoral degree (PhD and MBBS) and have received prior training in HC. One annotator also has clinical experience in delivering HC interventions. Prior to the annotation task, both annotators jointly reviewed relevant literature and established a shared understanding of the definitions and criteria for the S, M, and A components.

To minimize biases resulting from practice effects, they were presented with examples of well- and poorly formulated SMART goals. Additionally, a calibration exercise involving 10 sample goals, sourced independently of the main dataset, was conducted to ensure alignment and consistency in their interpretation and rating of goals. No reimbursement was given to the annotators as they were part of the research team.

C Extraction Prompt for LoRA Fine-Tuning of Generative Models

The prompt below defines the task used for fine-tuning generative models to extract goals from un-

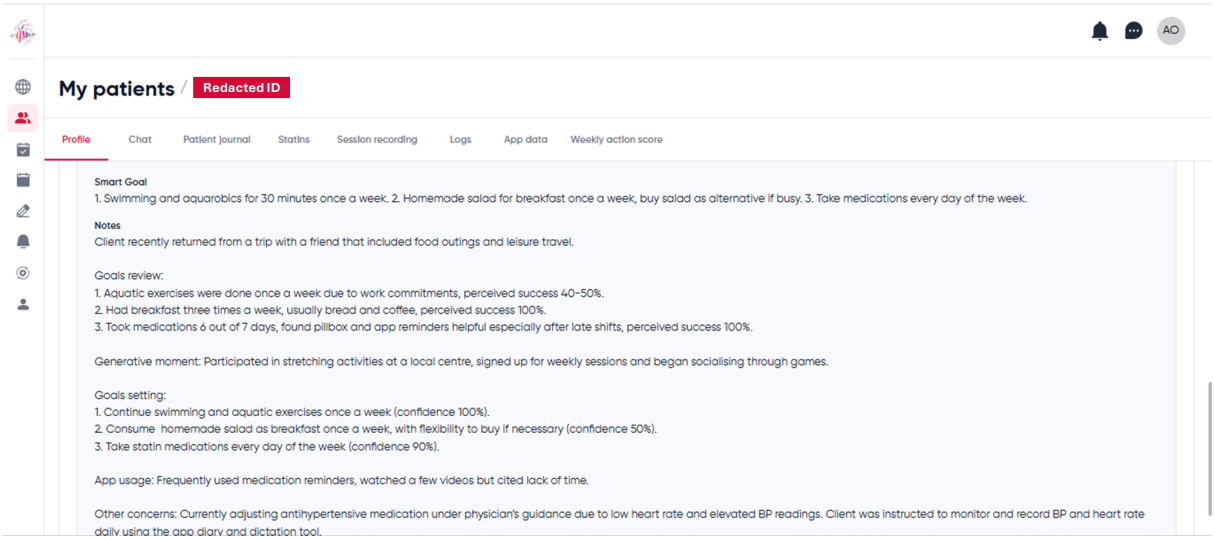

Figure 3: Example of a HC note recorded in the HC web-based platform.

structured HC notes. It clearly frames the task as extractive rather than generative, instructing the model to copy exact span text without paraphrasing. The instruction emphasizes exclusion of vague categories and long-term intentions, and focuses only on short-term, concrete weekly goals. The model is further guided to format the output as a numbered list.

```
Below is an instruction that describes a task, paired with an input
that provides further context. Write a response that appropriately
completes the request.

### Instruction:
You are an expert assistant that extracts only SMART weekly goals
from health coaching session notes. Extract the exact parts of
text, don't rephrase the text! This is an NLU task, and not an
NLG task! Only include goals that are: Specific, Measurable,
Attainable, Relevant, and Time-bound (SMART). Do not include
vague or broad categories like 'Exercise', 'Medication', or
'Diet' unless they are written as specific SMART goals. Ignore
6-month, long-term, or vague intentions. Focus only on short-term,
concrete weekly SMART goals that the patient committed to.
Format your response as a numbered list.

### Input: {input}

### Output: {output}
```

D Overview of Generative Models used for Fine-Tuning

Table 6 summarizes key details for the generative models used in our evaluation, including their release dates, pre-training cutoff dates, and the number of fine-tuned parameters relative to total model size. These models vary widely in scale and recency, which may influence their ability to extract structured SMART goals under zero-shot or fine-tuned conditions.

Model	Train params	Release	Cutoff
phi-4 (Abdin et al., 2024)	33M out of 14B	2024-12-13	Jun 2024
Mistral-Nemo-Instruct-2407	29M out of 12B	2024-07-18	Apr 2024
gemma-2-9b-it (Team et al., 2024)	27M out of 9B	2024-06-27	–
Meta-Llama-3.1-8B (Dubey et al., 2024)	21M out of 8B	2024-07-23	Dec 2023
mistral-7b-instruct-v0.3	21M out of 7B	2024/05/22	–
Llama-3.2-3B (Meta, 2024)	12M out of 3B	2024-09-25	Dec 2023

Table 6: Release and pre-training cutoff dates for used generative models.

E Zero-Shot SMART GPT-4.1 Goal Extraction Prompt and Schema

To support zero-shot structured extraction of weekly SMART goals, we used OpenAI's GPT-4.1 with the function-calling API. The model was instructed via a detailed system prompt (i.e., instruction) and constrained to return outputs in a predefined JSON schema. Below we provide both the prompt and the schema used.

Instruction

```
You are an expert assistant that extracts only SMART weekly goals
from health coaching session notes. Extract the exact parts of text,
don't rephrase the text! This is an NLU task, and not an NLG task!
Only include goals that are: Specific, Measurable, Attainable,
Relevant, and Time-bound (SMART). Do not include vague or broad
categories like "Exercise", "Medication", or "Diet" unless they
are written as specific SMART goals. Ignore 6-month, long-term,
or vague intentions. Focus only on short-term, concrete weekly
SMART goals that the patient committed to. Always respond in JSON
format.
```

Function Schema

```
[
  {
    "type": "function",
    "function": {
      "name": "extract_weekly_smart_goals",
      "description": "Extract only weekly SMART goals.
                      Ignore long-term or monthly goals.",
      "parameters": {
        "type": "object",
        "properties": {
          "goals": {
            "type": "array",
```

```
            "items": { "type": "string" },
            "description": "List of weekly SMART goals"
          }
        },
        "required": ["goals"]
      }
    }
  }
]
```

F Performance of the Goal Extraction Models Across SMARTSpan Splits

Table 7 reports the performance of `DeBERTa-v3-base` and `bert-base-uncased` on five test splits of the SMARTSpan dataset under three fine-tuning settings using the SpanQualifier framework: (*i*) trained only on SMARTSpan, (*ii*) first on MultiSpanQA then further fine-tuned on SMARTSpan, and (*iii*) trained only on MultiSpanQA. Results clearly show that models fine-tuned sequentially on MultiSpanQA and SMARTSpan achieve the highest scores across all splits, with `DeBERTa-v3-base` reaching up to **90.11** EM F1 and **97.41** PM F1, and `bert-base-uncased` up to **78.65** EM F1 and **89.62** PM F1. In contrast, models trained solely on SMARTSpan perform poorly, with EM F1 scores close to zero despite relatively higher PM recall. Models fine-tuned only on MultiSpanQA exhibit modest performance improvements over SMARTSpan-only training, but still fall short of the multi-stage fine-tuning setup. These findings underscore the importance of domain adaptation: while MultiSpanQA pretraining supports generalization, adaptation to SMARTSpan is essential for strong performance.

Table 8 reports the performance of `bert-base-uncased` and `roberta-base` across five test splits of the SMARTSpan dataset, using the CSS fine-tuning framework. Both models achieve strong and stable PM F1 performance, averaging above **85**. EM F1 scores for both models also vary within a similar range—approximately 65 to 81—suggesting comparable sensitivity to training data splits. While `roberta-base` achieves a slightly higher peak PM F1 score (**89.75**) and EM F1 score (**81.25**), `bert-base-uncased` performs competitively and exhibits a more balanced performance across metrics. These results suggest that both models are well-suited for multi-span extraction under CSS, with `roberta-base` offering marginally higher peaks and `bert-base-uncased` offering stable returns.

Table 9 shows that generative models fine-tuned with LoRA vary widely in their SMARTSpan performance. `mistral-7b-instruct-v0.3` emerges as the strongest overall, reaching up to **60.00** EM F1 and **90.34** PM F1 across splits. In contrast, larger models like `Mistral-Nemo-Instruct-2407` perform less consistently, peaking at only **53.52** EM F1 and **73.61** PM F1. Models such as `phi-4` and `Meta-Llama-3.1-8B` exhibit high variance, with EM F1 fluctuating from as low as **10.81** to **55.56**, indicating instability. Similarly, `gemma-2-9b-it` achieves strong PM scores (up to **77.62**) but lags in EM, highlighting challenges in exact span reproduction. Overall, smaller models with more stable tuning outperform larger counterparts on this task.

Table 10 presents the performance of the `flan-t5-large` model across five SMARTSpan test splits using the QASE framework. While PM F1 scores remain relatively stable, ranging from **51.21** to **62.93**, EM F1 scores are considerably

Model	EM			PM		
	P↑	R↑	F↑	P↑	R↑	F↑
DeBERTa-v3-base$_{\text{SMARTSpan_1}}$	0.26	19.57	0.52	21.74	81.29	34.31
DeBERTa-v3-base$_{\text{SMARTSpan_2}}$	0.35	21.57	0.70	27.84	78.34	41.08
DeBERTa-v3-base$_{\text{SMARTSpan_3}}$	0.12	12.24	0.23	21.05	73.05	32.69
DeBERTa-v3-base$_{\text{SMARTSpan_4}}$	0.11	10.42	0.22	20.83	68.63	31.96
DeBERTa-v3-base$_{\text{SMARTSpan_5}}$	0.58	38.89	1.14	19.56	77.64	31.24
DeBERTa-v3-base$_{\text{MultiSpanQA_SMARTSpan_1}}$	91.11	89.13	90.11	100.00	94.96	97.41
DeBERTa-v3-base$_{\text{MultiSpanQA_SMARTSpan_2}}$	81.13	84.31	82.69	88.03	90.55	89.27
DeBERTa-v3-base$_{\text{MultiSpanQA_SMARTSpan_3}}$	89.58	87.76	88.66	97.81	91.69	94.65
DeBERTa-v3-base$_{\text{MultiSpanQA_SMARTSpan_4}}$	78.26	75.00	76.60	91.44	83.91	87.51
DeBERTa-v3-base$_{\text{MultiSpanQA_SMARTSpan_5}}$	86.11	86.11	86.11	91.18	92.50	91.84
DeBERTa-v3-base$_{\text{MultiSpanQA_1}}$	11.11	10.87	10.99	40.08	23.36	29.52
DeBERTa-v3-base$_{\text{MultiSpanQA_2}}$	9.09	7.84	8.42	36.68	21.36	27.00
DeBERTa-v3-base$_{\text{MultiSpanQA_3}}$	6.00	6.12	6.06	32.94	18.91	24.03
DeBERTa-v3-base$_{\text{MultiSpanQA_4}}$	16.98	18.75	17.82	50.30	38.77	43.79
DeBERTa-v3-base$_{\text{MultiSpanQA_5}}$	19.51	22.22	20.78	37.75	34.82	36.22
bert-base-uncased$_{\text{SMARTSpan_1}}$	0.19	15.22	0.38	20.33	75.72	32.05
bert-base-uncased$_{\text{SMARTSpan_2}}$	0.42	25.49	0.83	26.13	87.35	40.23
bert-base-uncased$_{\text{SMARTSpan_3}}$	0.33	24.49	0.65	23.59	82.91	36.73
bert-base-uncased$_{\text{SMARTSpan_4}}$	0.24	20.83	0.48	24.43	80.42	37.48
bert-base-uncased$_{\text{SMARTSpan_5}}$	0.54	36.11	1.06	22.26	70.72	33.86
bert-base-uncased$_{\text{MultiSpanQA_SMARTSpan_1}}$	81.40	76.09	78.65	93.92	85.70	89.62
bert-base-uncased$_{\text{MultiSpanQA_SMARTSpan_2}}$	78.00	76.47	77.23	86.92	84.24	85.56
bert-base-uncased$_{\text{MultiSpanQA_SMARTSpan_3}}$	78.72	75.51	77.08	91.61	87.11	89.30
bert-base-uncased$_{\text{MultiSpanQA_SMARTSpan_4}}$	77.27	70.83	73.91	89.82	82.63	86.08
bert-base-uncased$_{\text{MultiSpanQA_SMARTSpan_5}}$	79.41	75.00	77.14	84.58	81.53	83.03
bert-base-uncased$_{\text{MultiSpanQA_1}}$	11.36	10.87	11.11	43.19	20.36	27.68
bert-base-uncased$_{\text{MultiSpanQA_2}}$	3.92	3.92	3.92	29.19	18.33	22.52
bert-base-uncased$_{\text{MultiSpanQA_3}}$	3.17	4.08	3.57	33.20	15.18	20.83
bert-base-uncased$_{\text{MultiSpanQA_4}}$	14.89	14.58	14.74	42.41	26.43	32.57
bert-base-uncased$_{\text{MultiSpanQA_5}}$	14.00	19.44	16.28	30.65	30.04	30.34

Table 7: Extractive models (`DeBERTa-v3-base` and `bert-base-uncased`) performance across five SMARTSpan test splits using SpanQualifier.

Model	EM			PM		
	P↑	R↑	F↑	P↑	R↑	F↑
roberta-base$_{\text{split_1}}$	78.00	84.78	81.25	89.81	87.77	88.78
roberta-base$_{\text{split_2}}$	52.38	86.27	65.19	66.71	93.20	77.76
roberta-base$_{\text{split_3}}$	69.70	93.88	80.00	84.10	96.20	89.75
roberta-base$_{\text{split_4}}$	55.13	89.58	68.25	71.88	95.18	81.90
roberta-base$_{\text{split_5}}$	73.81	86.11	79.49	83.69	91.67	87.50
bert-base-uncased$_{\text{split_1}}$	76.92	86.96	81.63	86.79	89.94	88.34
bert-base-uncased$_{\text{split_2}}$	53.75	84.31	65.65	71.53	90.63	79.95
bert-base-uncased$_{\text{split_3}}$	69.84	89.80	78.57	88.22	94.82	91.40
bert-base-uncased$_{\text{split_4}}$	69.49	85.42	76.64	88.52	89.49	89.00
bert-base-uncased$_{\text{split_5}}$	70.00	77.78	73.68	77.26	81.01	79.09

Table 8: Extractive models (`bert-base-uncased` and `roberta-base`) performance across five SMARTSpan test splits using CSS.

Model	EM			PM		
	P↑	R↑	F↑	P↑	R↑	F↑
phi-4$_{split_1}$	20.00	10.87	14.08	20.00	10.87	14.08
phi-4$_{split_2}$	24.14	13.73	17.50	36.19	21.04	26.61
phi-4$_{split_3}$	16.00	8.16	10.81	19.06	9.71	12.86
phi-4$_{split_4}$	47.22	35.42	40.48	65.52	48.49	55.73
phi-4$_{split_5}$	55.56	55.56	55.56	70.30	70.71	70.50
Mistral-Nemo-Instruct-2407$_{split_1}$	28.00	15.22	19.72	32.00	17.17	22.35
Mistral-Nemo-Instruct-2407$_{split_2}$	41.86	35.29	38.30	74.35	63.79	68.67
Mistral-Nemo-Instruct-2407$_{split_3}$	25.81	16.33	20.00	53.46	33.59	41.26
Mistral-Nemo-Instruct-2407$_{split_4}$	28.00	14.58	19.18	28.00	14.58	19.18
Mistral-Nemo-Instruct-2407$_{split_5}$	54.29	52.78	53.52	74.63	72.61	73.61
gemma-2-9b-it$_{split_1}$	37.50	39.13	38.30	74.74	75.95	75.34
gemma-2-9b-it$_{split_2}$	18.81	37.25	25.00	45.49	87.16	59.78
gemma-2-9b-it$_{split_3}$	23.53	16.33	19.28	50.13	36.64	42.33
gemma-2-9b-it$_{split_4}$	56.52	54.17	55.32	81.60	74.02	77.62
gemma-2-9b-it$_{split_5}$	46.88	41.67	44.12	62.95	56.20	59.38
Meta-Llama-3.1-8B$_{split_1}$	20.00	10.87	14.08	20.00	10.87	14.08
Meta-Llama-3.1-8B$_{split_2}$	32.43	23.53	27.27	51.51	37.80	43.60
Meta-Llama-3.1-8B$_{split_3}$	16.00	8.16	10.81	16.00	8.16	10.81
Meta-Llama-3.1-8B$_{split_4}$	26.92	14.58	18.92	34.38	18.56	24.11
Meta-Llama-3.1-8B$_{split_5}$	48.00	33.33	39.34	48.00	33.33	39.34
mistral-7b-instruct-v0.3$_{split_1}$	34.15	30.43	32.18	74.09	65.85	69.73
mistral-7b-instruct-v0.3$_{split_2}$	42.50	33.33	37.36	63.42	49.55	55.63
mistral-7b-instruct-v0.3$_{split_3}$	45.00	55.10	49.54	77.26	90.71	83.45
mistral-7b-instruct-v0.3$_{split_4}$	57.69	62.50	60.00	88.54	92.21	90.34
mistral-7b-instruct-v0.3$_{split_5}$	52.17	66.67	58.54	70.94	89.15	79.01
Llama-3.2-3B$_{split_1}$	19.57	19.57	19.57	47.20	42.68	44.83
Llama-3.2-3B$_{split_2}$	26.32	39.22	31.50	41.96	53.81	47.15
Llama-3.2-3B$_{split_3}$	22.00	22.45	22.22	43.41	39.80	41.53
Llama-3.2-3B$_{split_4}$	35.19	39.58	37.25	57.70	64.44	60.88
Llama-3.2-3B$_{split_5}$	19.54	47.22	27.64	26.38	58.63	36.39
flan-t5-large$_{split_1}$	51.35	41.30	45.78	72.03	76.99	74.43
flan-t5-large$_{split_2}$	55.56	39.22	45.98	75.92	78.65	77.26
flan-t5-large$_{split_3}$	25.49	26.53	26.00	50.33	70.86	58.86
flan-t5-large$_{split_4}$	47.37	37.50	41.86	77.45	82.81	80.04
flan-t5-large$_{split_5}$	42.11	44.44	43.24	56.97	81.59	67.09
bart-large$_{split_1}$	40.62	28.26	33.33	83.10	64.63	72.71
bart-large$_{split_2}$	15.62	9.80	12.05	49.75	52.96	51.30
bart-large$_{split_3}$	30.00	18.37	22.78	67.84	48.88	56.82
bart-large$_{split_4}$	45.45	31.25	37.04	81.82	61.84	70.44
bart-large$_{split_5}$	42.31	30.56	35.48	66.44	50.24	57.22

Table 9: Generative models performance across five SMARTSpan test splits using LORA fine-tuning.

lower, with values between **26.67** and **37.04**. This pattern highlights the model's ability to identify semantically relevant spans but also underscores the challenge of achieving exact span boundary matches. The discrepancy between PM and EM metrics reflects the difficulty in generating strictly accurate extractions under the QASE setup.

Model	EM			PM		
	P↑	R↑	F↑	P↑	R↑	F↑
flan-t5-large$_{split_1}$	42.86	32.61	37.04	78.53	52.50	62.93
flan-t5-large$_{split_2}$	30.77	23.53	26.67	70.88	50.00	58.63
flan-t5-large$_{split_3}$	35.00	28.57	31.46	73.68	50.64	60.03
flan-t5-large$_{split_4}$	40.00	29.17	33.73	73.52	49.12	58.89
flan-t5-large$_{split_5}$	30.56	30.56	30.56	55.14	47.80	51.21

Table 10: `flan-t5-large` model performance across five SMARTSpan test splits using QASE.

Table 11 reports the performance of `GPT-4.1` across five SMARTSpan test splits. The model achieves strong PM F1 scores, ranging from **69.74** to **84.94**, indicating high semantic alignment with relevant spans. However, EM F1 scores are substantially lower, varying between **23.26** and **44.78**, highlighting challenges in predicting exact span boundaries. These results suggest that while `GPT-4.1` is highly effective at identifying relevant content in zero-shot settings, it falls short in producing precise extractions. The consistent PM performance nonetheless underscores its utility for approximate span retrieval tasks.

Model	EM			PM		
	P↑	R↑	F↑	P↑	R↑	F↑
GPT-4.1$_{split_1}$	48.72	41.30	44.71	96.55	75.83	84.94
GPT-4.1$_{split_2}$	33.33	27.45	30.11	84.46	61.26	71.01
GPT-4.1$_{split_3}$	27.03	20.41	23.26	83.01	60.13	69.74
GPT-4.1$_{split_4}$	42.00	43.75	42.86	79.60	71.79	75.49
GPT-4.1$_{split_5}$	48.39	41.67	44.78	79.22	68.07	73.22

Table 11: `GPT-4.1` models performance across five SMARTSpan test splits using LLM Schema.

G Performance of the SMARTness Classification Models Across SMARTSpan Splits

Across the five SMARTSpan test splits, both `deberta-v3-base` and `deberta-v3-large` exhibit consistently strong performance in SMARTness classification, with macro F1 scores exceeding 0.740 and SMART F1 scores ranging from 0.864 to 0.955. `deberta-v3-large` achieves the highest overall scores, reaching up to **0.875** accuracy, **0.881** macro F1, and a peak SMART F1 of **0.955**, demonstrating that increased model capacity leads to greater robustness and precision. In contrast, while `roberta-large` performs competitively on most splits, it shows instability on Split_ 3, where classification performance collapses entirely (SMART F1 = **0.000**). This divergence highlights the importance of evaluating models across multiple test partitions and supports the reliability of DeBERTa-based architectures—particularly the large variant—for structured SMARTness prediction in goal-oriented health coaching contexts.

H Output for `DeBERTa-v3-base` fine-tuned only on SMARTSpan using SpanQualifier

The list below shows the final inference output of `DeBERTa-v3-base` trained with the SpanQualifier

Model	Accuracy↑	Macro F1↑	SMART F1↑
deberta-v3-base$_{split_1}$	0.780	0.744	0.900
deberta-v3-base$_{split_2}$	0.809	0.786	0.864
deberta-v3-base$_{split_3}$	0.822	0.782	0.913
deberta-v3-base$_{split_4}$	0.902	0.897	0.927
deberta-v3-base$_{split_5}$	0.833	0.836	0.870
deberta-v3-large$_{split_1}$	0.854	0.842	0.900
deberta-v3-large$_{split_2}$	0.830	0.809	0.864
deberta-v3-large$_{split_3}$	0.867	0.834	0.955
deberta-v3-large$_{split_4}$	0.878	0.881	0.895
deberta-v3-large$_{split_5}$	0.875	0.874	0.909
roberta-large$_{split_1}$	0.780	0.743	0.923
roberta-large$_{split_2}$	0.787	0.745	0.851
roberta-large$_{split_3}$	0.244	0.131	0.000
roberta-large$_{split_4}$	0.878	0.860	0.913
roberta-large$_{split_5}$	0.833	0.849	0.833

Table 12: Encoder-only transformer models (`deberta-v3-base`, `deberta-v3-large` and `roberta-base`) performance across five SMARTSpan test splits.

framework on the SMARTSpan dataset using only 123 training examples. This model achieved an average EM F1 of **0.56** and average PM F1 of **34.26**. As shown below, the model returns a large set of overlapping span candidates with minimal filtering or boundary control.

This output illustrates SpanQualifier's reliance on large-scale supervision. In low-resource conditions, it fails to discriminate meaningful spans from irrelevant ones and defaults to producing dense n-gram windows. Without upstream training (e.g., on MultiSpanQA) or parameter-efficient adaptation (e.g., LoRA), the model lacks a coherent span representation and performs poorly on SMARTSpan.


```
"24": [
    ") exercise : switch from casual walking to brisk walking",
    "exercise : switch from casual walking to brisk walking ,",
   ": switch from casual walking to brisk walking , maintaining",
   "switch from casual walking to brisk walking , maintaining a",
    "from casual walking to brisk walking , maintaining a daily",
    "casual walking to brisk walking , maintaining a daily goal",
    "walking to brisk walking , maintaining a daily goal of",
    "to brisk walking , maintaining a daily goal of 10",
    "brisk walking , maintaining a daily goal of 10 ,",
    "walking , maintaining a daily goal of 10 , 000",
    ", maintaining a daily goal of 10 , 000 steps",
    "maintaining a daily goal of 10 , 000 steps (",
    "a daily goal of 10 , 000 steps ( confidence",
    "daily goal of 10 , 000 steps ( confidence 7",
    "goal of 10 , 000 steps ( confidence 7 /",
    "of 10 , 000 steps ( confidence 7 / 10",
    "10 , 000 steps ( confidence 7 / 10 )",
    ", 000 steps ( confidence 7 / 10 ) .",
    "000 steps ( confidence 7 / 10 ) . (",
    "steps ( confidence 7 / 10 ) . ( 2",
    "( confidence 7 / 10 ) . ( 2 )",
    "confidence 7 / 10 ) . ( 2 ) diet",
    "7 / 10 ) . ( 2 ) diet :",
    "/ 10 ) . ( 2 ) diet : reduce",
    "10 ) . ( 2 ) diet : reduce the",
    ...
    "activity . future sessions may explore strategies to
    support consistent medication intake and improving sleep"
    ]
```


I Output for `DeBERTa-v3-base` after Sequential Fine-Tuning on MULTISPANQA and SMARTSPAN using SpanQualifier

The list below presents the final output of `DeBERTa-v3-base` fine-tuned using the SpanQualifier framework, first on MultiSpanQA and subsequently on SMARTSpan. Unlike the low-resource setting where the model was trained only on SMARTSpan, this model produces a concise and accurate set of spans, closely aligned with the gold annotations. The final predictions are well-formed and exhibit clear span boundaries, indicating that prior training on a large-scale multi-span dataset (i.e., MultiSpanQA) successfully bootstraps the model's ability to extract meaningful multi-span goals from SMARTSpan.

This result underscores the importance of task-aligned pretraining and sufficient supervision. By first training on a diverse, high-resource dataset and then adapting to the smaller in-domain SMARTSpan, the model effectively generalizes and avoids overgeneration.


```
"24": [
  "switch from casual walking to brisk walking, maintaining
  a daily goal of 10, 000 steps (confidence 7/10)",
  "reduce the overall quantity of food consumed by 25 %,
  confidence 7/10",
  "reduce blood sugar"
  ]
```


J Output for `Mistral-Nemo-Instruct-2407` Fine-Tuned on MultiSpanQA

Examples of HTML tags:


```
"m20r1ol7e07go0pqexqq": [
  "</div >",
  "France",
  "Galician",
  "Germany",
  "Italy",
  "Portugal",
  "Spain",
  "UK",
  "the Americas"
],

"1ove66khudx8x16s6i81": [
  "</div>",
  "<br />",
  "<div class=\"column col-3 col-last\">",
  "south of Vancouver , British Columbia",
  "the Hell on Wheels set",
  "the University Women 's Club of Vancouver"
]
```


Example of stray characters:


```
"v9ohyt3b06g4y03cqysk": [
  ")",
  "CANOE",
  "OCEAN",
  "activity",
  "agreeableness",
  "assertiveness",
  "conscientiousness",
```

```
    "excitement seeking",
    "extraversion",
    "gregariousness",
    "neuroticism",
    "openness to experience",
    "positive emotions",
    "warmth"
  ],

  "hblv00fzqtwavy96o582": [
    ":",
    "Don Felder",
    "Don Henley",
    "Glenn Frey",
    "Joe Walsh",
    "the Eagles",
    "the band"
  ]
```

Other examples:

```
  "503kdfmuelcn8in1eggi": [
    "### Input:",
   "Ansel Elgort ( born March 13, 1994 ) is an American actor and
    musician. He is best known for his leading role in the 2014
   film The Fault in Our Stars, and for playing the title character
    in the 2017 film Baby Driver. He has also appeared in the
    films Divergent ( 2014 ), The Divergent Series: Insurgent
    ( 2015 ), and The Divergent Series: Allegiant ( 2016 ), and
    in the 2017 film The Black Balloon. Elgort was born in New
    York City, the son of Arthur Elgort, a fashion photographer,
    and Grethe Barrett Holby, an artist. His father is Jewish,
    while his mother is of Norwegian and English descent. He
    was named after his great - grandfather, the Swedish - born
   artist Andrew Olsson. He grew up in Manhattan and attended the
   Professional Children 's School. He began acting at age 9, and
   began taking acting classes at the Stagedoor Manor summer camp.
    He made his stage debut in the 2002 production of the musical
    Guys and Dolls at the Kennedy Center in Washington, D.C. He
    appeared in a short film, The Scapegoat, in 2002, and in the
    film Daddy 's Home in",
    "Edgar Wright",
    "Eiza González",
    "Jamie Foxx",
    "Jon Bernthal",
    "Jon Hamm",
    "Kevin Spacey",
    "Lily James"
  ]
```

K Output for `Llama-3.2-3B` Fine-Tuned on SMARTSpan

Example of made up content:

```
Input: "final session. at the request of the client, the session was
short and focused on summarizing past experience and suggestions for
improvement. client reflects a desire to reduce cholesterol levels
and medication intake and had hoped for an intervention with immediate
 results. suggests that health coaching sessions could be more
directive and structured. finds the educational materials useful
but did not access them, stating they felt like \"common sense\"
 and expressing a general preference for not using the phone
frequently. describes managing diet and exercise in a strict way
 and felt demoralized when cholesterol levels did not improve.
 hc reflected that small changes were still observed, such as
 greater variety in diet. client shared that \"resistance is
futile\" and accepts the need for long-term medication. hc affirmed
that lifestyle may be helping prevent further worsening despite
lack of measurable improvement. client reported bugs in the app:
receiving multiple medication reminders one hour after the original,
and not receiving notifications while overseas even though login
was possible. hc closed the session with well wishes and encouragement
 to continue prioritizing health."

 Output: [
    "explore ways to improve diet and exercise",
    "explore ways to improve health literacy",
    "explore ways to improve medication adherence",
    "explore ways to improve overall health",
    "explore ways to improve sleep",
    "explore ways to increase medication adherence",
    "explore ways to reduce cholesterol levels",
    "explore ways to reduce medication intake",
    "explore ways to reduce stress"
  ]
```