From Dialogue to Data:

Leveraging Large Language Models to Streamline Survey Data Collection in Social Work



Shangchanhui Feng

EC Utbildning

Examensarbete

202508

# Abstract

This thesis presents the design, implementation, and evaluation of a minimum viable product (MVP) for an AI-assisted survey tool aimed at improving data collection processes in social work case management. The tool integrates speech-to-text transcription via OpenAI's Whisper and language understanding through GPT-4o-mini to automatically populate structured survey responses from conversational interviews. Key features include real-time progress display, AI confidence flagging, and a human-in-the-loop review system. A technical evaluation was conducted to explore trade-offs between accuracy, latency, and cost using different chunking configurations. Results indicate that moderate chunk sizes with 2-sentence overlap optimize accuracy while maintaining acceptable processing speed and resource use. The thesis also outlines future development directions, including real-time interaction, integration of domain-specific knowledge, offline capabilities, and compliance with data protection regulations. This work demonstrates the potential for generative AI to support more efficient, scalable, and human-centered data practices in social services and beyond.

# Acknowledgement

I would like to begin by thanking my teacher, **Antonio Prgomet**, for introducing me to the world of data science. His insight, guidance, and encouragement have been invaluable throughout this journey.

A heartfelt thank you to my employer, **SOS Children’s Villages Sweden**, for generously supporting my studies and granting me the flexibility to learn while working. I'm particularly grateful for the opportunity to immediately apply my knowledge and test new ideas in real-world settings. Without the support of SOS, I would not be where I am today.

My gratitude also goes to my former colleagues at SOS, **Hanna-Liisa Juntti** and **Magda Skyllbäck**, experts in social work, who kindly recorded realistic mock interviews. These recordings were crucial for building and testing the prototype.

Many thanks go to my colleagues at SOS international office, **Dario Peter** and **Florian Edelmaier**, for your continuous interest, unwavering support, and belief in my work. Collaborating with you over the past year has been a highlight of my professional life. You have been the strongest advocates for my ideas, mobilizing resources and creating space for innovation. Your genuine passion and strong work ethic continue to deeply inspire me.

Last but not least, special thanks to my partner **Fredrik Carlsson**, for his unconditional emotional and technical support. His dedication to AI and science has been my greatest source of inspiration throughout this journey.

Table of Contents

[Abstract 2](#_Toc204944929)

[Acknowledgement 3](#_Toc204944930)

[1 Introduction 1](#_Toc204944931)

[2 Theory 2](#_Toc204944932)

[2.1 Social Work Case Management in Practice 2](#_Toc204944933)

[2.2 Large Language Models (LLMs) 3](#_Toc204944934)

[2.2.1 Transformer Architecture and Attention Mechanisms 3](#_Toc204944935)

[2.2.2 Reasoning models 3](#_Toc204944936)

[2.2.3 Context window 4](#_Toc204944937)

[2.2.4 Prompt engineering 4](#_Toc204944938)

[2.3 Performance optimisation 4](#_Toc204944939)

[2.3.1 Accuracy 5](#_Toc204944940)

[2.3.2 Latency 5](#_Toc204944941)

[2.4 Evaluation Metrics 6](#_Toc204944942)

[2.4.1 Accuracy metrics 6](#_Toc204944943)

[2.4.2 Latency and cost metrics 6](#_Toc204944944)

[3 Method 7](#_Toc204944945)

[3.1 Frontend interface 7](#_Toc204944946)

[3.2 Data pre-processing 7](#_Toc204944947)

[3.2.1 Audio data 7](#_Toc204944948)

[3.2.2 Survey data 8](#_Toc204944949)

[3.3 Pipeline and key features 9](#_Toc204944950)

[3.3.1 LLM Prompt Design and Output Handling 9](#_Toc204944951)

[3.3.2 Idempotent Update & data storage 9](#_Toc204944952)

[3.3.3 Transparency and certainty flagging 10](#_Toc204944953)

[3.3.4 Human-in-the-loop & overrides 10](#_Toc204944954)

[3.4 Evaluation 10](#_Toc204944955)

[4 Result and Discussion 11](#_Toc204944956)

[4.1 Evaluation result 11](#_Toc204944957)

[4.2 Future outlook 13](#_Toc204944958)

[4.2.1 Real-time transcription and interaction 13](#_Toc204944959)

[4.2.2 Agentic AI with Domain Knowledge 13](#_Toc204944960)

[4.2.3 Technology Accessibility 13](#_Toc204944961)

[4.2.4 Sensitive Data Handling and Regulatory Compliance 13](#_Toc204944962)

[5 Conclusion 14](#_Toc204944963)

[Appendix A example prompt 15](#_Toc204944964)

[Appendix B example survey 16](#_Toc204944965)

# Introduction

Surveys are essential tools in social work for capturing the needs of children, youth, and families and for measuring the effectiveness of interventions. Surveys consistently provide critical information for assessing program impacts and help organizations adapt their efforts to better meet each individual's unique needs. Furthermore, reliable survey data serve as a strong foundation for demonstrating success to donors, stakeholders, and the broader community, fostering robust partnerships and successful fundraising.

In Sweden, social workers typically facilitate surveys digitally while engaging participants in dialogue. In many other countries, organizations such as SOS Children's Villages (hereafter SOS) rely on offline apps (on phones or tablets) linked to a central system, or paper surveys that must later be manually entered into databases. Regardless of the specific method, these processes share common pain points:

* **Repetitive, manual, and time-consuming processes**, often resulting in incomplete data.
* **Interruptions in conversational flow**, affecting the natural interaction between social workers and participants.
* **Participant discomfort and fatigue**, especially during lengthy questionnaires, leading to lower engagement and incomplete responses.
* **Inconsistent data quality** due to variations in standards, interpretations, and capacities among social workers.

These challenges lead to delayed insights, compromised productivity, and inconsistent data quality, ultimately hindering the effectiveness and impact of social programs. In 2024 alone, over 73,000 children and youth participated in the SOS alternative care programs, and more than 98,000 households were involved in family-strengthening programs, each undergoing survey processes one to two times annually. The issues outlined are widespread among NGOs and government social services worldwide, suggesting a clear need for scalable solutions that streamline survey processes to enhance productivity and impact across public and civil society sectors.

This project addresses these challenges by exploring the development of an AI-assisted solution to streamline survey completion. The proposed approach involves creating a prototype (minimum viable product) that allows user to upload a survey (in designated format), then leverages advanced speech-to-text transcription technologies such as **OpenAI’s Whisper**, and a large language model (e.g. **GPT-o4 mini**) to analyse transcriptions and automatically populate structured survey responses in near real-time. A user-friendly interface will flag AI-generated answers that have higher uncertainty, enabling social workers to easily review and edit automatically filled surveys, preserving essential human oversight and ensuring data accuracy. The completed surveys will be downloadable in structured formats.

The purpose of this project is to evaluate whether an AI-driven survey assistant can enhance efficiency in survey administration while ensuring data quality and preserving the vital human element in social work case management. To fulfill this purpose, the following research questions will be investigated:

1. **How effectively can an AI-driven system transcribe and automatically extract structured survey information from conversational interviews?**
2. **How do various text-chunking strategies (size and overlap) affect the accuracy and computational efficiency of the AI-assisted survey completion process?**

By answering these questions, the project aims to provide valuable insights into the technical performance of the AI system, optimize text-processing techniques, and address broader implementation considerations, ultimately enhancing the effectiveness of survey processes for both social work practitioners and participants

All code for the prototype and evaluation presented in this thesis is available at: <https://github.com/AmyFeng0227/AI-assisted-survey-tool>

# Theory

## Social Work Case Management in Practice

In the context of SOS, social work case management is a structured process designed to provide tailored support to children, youth, and families. It typically involves survey completion at different stages: initial assessment (inskrivning), follow-up assessments (ideally every six months), and case closure (utskrivning). At each stage, data collection plays a central role—not only to determine the kind of support a participant should receive but also to monitor progress over time and assess intervention effectiveness at both individual and community levels. Surveys are the primary tools for this data collection.

A typical survey begins with basic personal information (e.g., age, gender). Personal Identifiable Information (PII) such as names is anonymized using participant IDs to ensure data protection. Surveys often include thematic sections covering key life areas such as accommodation, education and employment, social networks, and physical and psychological health. Question types vary and include:

* **Single-choice questions** (e.g., “Do you have stable housing?” with answers like Yes/No or on a 5-point scale from “Strongly agree” to “Strongly disagree”)
* **Multiple-choice questions** (e.g., “What are your sources of income?”, with answer options like study loan/work/financial assistance, etc.)
* **Open-ended text fields** (e.g., “What kind of support do you want?”)

Many surveys also provide optional note fields for each question, enabling social workers to record context-specific details that are crucial for ongoing case planning.

In Sweden, surveys are usually completed using an online system, with social workers and youth reviewing questions together on a shared screen. In some other global SOS locations, the process may differ depending on the work context: paper-based forms and offline mobile/tablet apps are used to collect responses for subsequent digital upload, particularly in communities where internet access is unreliable or unavailable. Regardless of format, the process is time-consuming and often disrupts conversational flow, highlighting a critical opportunity for innovation—leveraging AI to streamline and enhance data collection while preserving human connection and professional judgment.

Despite its importance, the current approach to survey-based data collection faces some challenges. First, there is the issue of high workload. Ideally, social workers should manage no more than 10–20 individual cases simultaneously, depending on case complexity. However, due to limited staffing and resources, caseloads can exceed this recommendation, compelling social workers to prioritize direct participant engagement over administrative tasks such as survey completion. This trade-off often results in incomplete or delayed data entry.

Second, the problem of unstandardised answers arises from diverse educational backgrounds and professional experiences among social workers, leading to inconsistencies in interpreting and recording survey responses. For instance, two social workers might assess the severity of a youth’s psychological distress differently despite encountering similar narratives. This variability reduces data comparability across individuals, cultures, and countries, complicating global aggregation efforts. Introducing AI-assisted tools capable of applying consistent reasoning logic—especially when trained on domain-specific knowledge—offers potential to reduce administrative burdens and response variability, thereby improving the efficiency and quality of social work case documentation.

## Large Language Models (LLMs)

In recent years, generative AI has advanced significantly, laying the foundation for sophisticated automation in language tasks. A pivotal milestone was the release of OpenAI’s ChatGPT in late 2022, showcasing the ability of large language models (LLMs) to engage in fluent dialogue and driving mainstream adoption. Shortly thereafter, OpenAI introduced GPT-4 in 2023, a model with significantly enhanced capabilities, achieving near-human-level performance on numerous academic and professional benchmarks. For instance, GPT-4 can pass a simulated bar exam in the top 10% of test-takers, whereas the earlier GPT-3.5 scored in the bottom 10% (OpenAI, 2023). Such breakthroughs illustrate that contemporary generative models can manage complex comprehension and generation tasks previously unattainable through automation. This leap in capability—combined with the widespread visibility of systems like ChatGPT—has sparked rapid growth in large language model development and increasing interest in applying these models to practical, real-world problems, including the challenges addressed by this project.

### Transformer Architecture and Attention Mechanisms

Transformers are a neural network architecture introduced by Vaswani et al. (2017) in their seminal paper “Attention is all you need,” and have since become the backbone of state-of-the-art models driving modern generative AI. Attention mechanisms help deep learning models prioritize the most relevant parts of input data. This innovation enabled the Transformer architecture, which led to today’s LLMs, such as ChatGPT. Transformers were developed to overcome limitations of earlier models like recurrent neural networks (RNNs), which process text sequentially and struggle with long-term dependencies.

Inspired by human cognitive focus, attention mechanisms allow Transformers to relate information across any distance within the input sequence, effectively capturing long-range dependencies. In practice, attention computes weights representing the importance of each input segment, enabling the model to emphasize relevant data selectively (IBM, n.d.).

Because of their speed, scalability, and accuracy, Transformers underpin technologies like OpenAI's Whisper (speech-to-text transcription) and GPT models (language understanding), both employed in this project to automate survey completion from social work interviews.

### Reasoning models

Reasoning models, such as GPT-4-mini used in this project, are trained through reinforcement learning to perform complex reasoning tasks. Unlike traditional models that generate outputs directly, reasoning models engage in a "private chain of thought"—an internal process where the model explores multiple solution strategies before producing a final response.

These models are particularly effective in complex problem-solving, scientific reasoning, coding, and multi-stage planning, and are often employed in agentic workflows involving tool use. A key feature is the inclusion of special reasoning tokens alongside standard input and output tokens. These tokens allow the model to internally analyze the prompt and reason through different approaches.

After completing this internal reasoning, the model outputs the final result as visible completion tokens, while the reasoning tokens are discarded from the returned context. Although not exposed via the API, reasoning tokens still occupy space in the model's context window and are billed as output tokens.

Therefore, when using reasoning models, it is especially important to ensure a large context window, allowing sufficient space for both internal reasoning and output generation. One should also be aware of the relatively higher token cost.

### Context window

Another key feature of Transformer-based LLMs is their context window—the number of tokens a model can process in a single input. Early generative models, such as GPT-2, were limited to just a few hundred tokens. In contrast, modern models support vastly larger context lengths. For instance, GPT-4-mini, used in this project, supports a context window of 200,000 tokens (OpenAI, 2025), while Gemini 1.5 Pro can handle up to 2 million tokens (Kilpatrick et al., 2024).

This substantial increase in context size has revolutionized how LLMs are used, enabling a broader range of applications and new implementation strategies. However, context size presents a trade-off: while larger windows allow the model to access more information at once, they also increase computational load and may dilute the model’s focus if irrelevant content is included.

To address this, we segmented inputs when necessary and ensured each query remained within the token limits of the selected model, thereby optimizing both performance and cost-efficiency.

### Prompt engineering

Because LLMs are general-purpose text generators by default, producing precise and structured outputs requires careful prompt engineering. Prompt engineering refers to the practice of crafting the input text (i.e., the prompt) in a way that guides the model toward the desired behaviour and output format.

In the context of a survey tool, models do not inherently understand multiple-choice logic or form-filling conventions. Instead, structured output can be achieved through natural language instructions and illustrative examples. For instance, a prompt might specify: “Answers to single- or multiple-choice questions should be in a list. Output only for the questions that are clearly addressed in the transcript.” Including a few-shot example can further enhance output consistency.

In many cases, prompt engineering is the first—and sometimes only—step required to attain strong model performance on specific tasks. Reasoning models tend to respond well to broad, high-level instructions, while standard GPT-style models typically perform better when given explicit, step-by-step directions (OpenAI, 2025).

As demonstrated in this project, prompt engineering plays a crucial role in adapting general-purpose models for structured, domain-specific applications such as social work survey automation.

## Performance optimisation

Deploying LLM-powered systems often involves a trade-off between accuracy and latency. Generally, achieving higher accuracy or more advanced model capabilities comes at the cost of slower response times and increased computational expense. This project had to balance these considerations to deliver a tool that is both reliable in extracting accurate survey responses and fast enough for use in field settings.

### Accuracy

The definition of accuracy depends on the specific use case. For a survey tool, satisfactory accuracy means that the LLM produces results that significantly reduce manual effort while tolerating a small number of mistakes or uncertainties—issues that are resolved through a human-in-the-loop process.

According to OpenAI (2025), there are two main strategies for improving LLM output accuracy: context optimisation and LLM optimisation.

* **Context optimisation** enhances the model's response quality by providing more relevant, domain-specific input—especially when required knowledge is missing from training data. Techniques include prompt engineering, retrieval-augmented generation (RAG), and fact-checking mechanisms.
* **LLM optimisation** focuses on improving model behaviour in terms of formatting, tone, and reasoning. This can be addressed through prompt design and, where necessary, fine-tuning.

These approaches are not mutually exclusive and can be combined and iteratively refined. In this project, acceptable accuracy was achieved through prompt engineering alone. However, further improvements could be explored via RAG or fine-tuning in future iterations, especially for more complex tasks requiring deeper domain knowledge (e.g., specialized thematic surveys).

### Latency

Latency refers to the time between submitting a prompt and receiving the model's response. It directly affects user experience, particularly in applications aiming for near-real-time interaction.

Several strategies can improve latency:

* **Model selection**: Larger models tend to have slower inference times. Reasoning models, which perform internal chains of thought, are slower by nature. This project chose GPT-4-mini to balance higher accuracy with lighter-weight architecture, reducing both latency and token cost.
* **Token management**: Output token length has a greater effect on latency than input token length. For example, reducing output tokens by 50% may reduce latency by 50%, while cutting input tokens by 50% may reduce latency by only 1–5% (OpenAI, 2025).
* **Prompt compression**: An early experiment in this project used **semantic search with vector embeddings** to select only the most relevant survey questions, rather than submitting the entire survey to the model. This aimed to improve both accuracy and latency by narrowing the input context. However, the results showed unsatisfactory selection accuracy. Given the rapid growth of context windows in most LLMs and the limited latency gain from reduced input size, this approach was not adopted for the MVP.
* **Chunking**: To simulate real-time progress, the MVP processes long audio files by first transcribing them into text, then dividing the transcript into small chunks (a few sentences each). These chunks are sent to the LLM sequentially, allowing the user to see progressively updated survey responses. It is however with noticing that smaller chunks yield more frequent updates and create the perception of shorter latency, although the total latency on the same text may take much longer.

This reflects a classic **accuracy vs. latency trade-off**, which will be further addressed in the next section through a proposed evaluation framework to determine the optimal balance for different use cases.

## Evaluation Metrics

To assess whether the prototype fulfills its core purpose—reliably completing survey responses while remaining fast and cost-effective—two complementary groups of evaluation metrics are defined.

### Accuracy metrics

Accuracy metrics measure the proportion of questions that the system handles correctly. This evaluation considers:

* **True positives (TP):** questions with correctly filled answers
* **True negatives (TN):** questions rightly left blank.
* **False negatives (FN)** capture questions that the model failed to answer even though the correct information was present in the transcript.
* **False positives (FP)** are divided into two types:
  + **FP\_W (Wrong)**: The model provided an incorrect answer.
  + **FP\_U (Unwanted)**: The model answered a question that should have been left blank (i.e., hallucinated an answer).

An overall **accuracy rate** is calculated as: **(TP + TN) / Total number of questions**

### Latency and cost metrics

This group of metrics captures the practical trade-offs of achieving accuracy: response time, consistency, and resource usage.

* **Round-Trip Time (RTT)** measures the time between sending a chunk of text to the language model API and receiving a complete parsed response. It serves as a direct proxy for per-chunk latency. Because occasional network delays or model slowdowns may skew results, this project uses the trimmed mean RTT—discarding the fastest and slowest 10% of calls to calculate a more robust average. Multiplying this trimmed mean RTT by the number of chunks provides an estimate of the end-to-end runtime.
* **Retry Count**: Some API calls fail and must be retried. Each retry adds to the total latency and token usage, and therefore is included in the evaluation.
* **Total Tokens Used**: As token-based APIs bill based on input and output length, the total token count is used as a direct proxy for monetary cost.

Together, these metrics expose the classic trade-off: Pushing for higher accuracy—e.g., by using smaller, overlapping chunks—increases the number of tokens, raises the total number of chunks, and heightens the risk of retries. This inflates both latency and cost. By presenting latency and cost metrics alongside accuracy, it becomes clear that the "best" configuration is not simply the most correct—but the one that delivers acceptable accuracy within the constraints of time, budget, and operational workflow.

# Method

This section outlines the methodology used to implement a minimum viable product (MVP) of an AI-assisted survey tool designed for social work case management. The finalized MVP is presented as a Streamlit application and includes the following functionalities:

* Upload of survey templates
* Upload and transcription of audio files using OpenAI’s Whisper
* AI-assisted answering of survey questions with near real-time updates using GPT-4o-mini
* Confidence flagging of AI-generated answers
* Human editing capability, with question locking to prevent further AI modifications
* A visual progress bar indicating completion status
* A button to download the completed survey

Additionally, an evaluation system was built to assess the product’s latency and accuracy.

Below is a breakdown of the key technologies used and the design principles applied.

## Frontend interface

The prototype’s user interface is implemented using **Streamlit**, a Python web app framework that enables rapid development of interactive dashboards. Custom CSS, inspired by **Tailwind utility classes**, is applied to support color-coding and clean layout design for a streamlined user experience.

Upon launching the application, users are prompted to upload two files:

1. A survey template in Excel format
2. One or more audio recordings of interviews

After processing, a progress bar shows how many survey questions have been answered.

Below the bar, survey questions are displayed in two columns:

* Left: Answered questions
* Right: Unanswered questions

Each question is color-coded to indicate its status:

* Green / Yellow / Red for high / medium / low confidence
* Grey for unanswered
* Highlighted green for human-edited entries

The interface also includes buttons to:

* Clear the cache and restart the application
* Download the completed survey as an Excel file

## Data pre-processing

### Audio data

Ideally, the final product would include a live speech-to-text feature that transcribes ongoing conversations in real time. As the audio is transcribed, the text would be chunked on-the-fly and sent to the language model for prompt-based survey completion—thereby enabling a near-live user experience. However, due to the technical complexity and the scope limitations of an MVP, this functionality was substituted with an audio upload feature. Users can upload audio files in MP3 or M4A format (sample files are typically 5–15 minutes long). The uploaded audio is transcribed as a whole using OpenAI’s Whisper model, chosen for its high accuracy, strong noise handling, and broad language support. This MVP is currently tested with English only.

Once transcribed, the full text is chunked into smaller groups all at once to mimic a near-real-time experience. This logic is implemented using an idempotent processing flow as preparation for future real-time functionality.

Chunking follows two main parameters: Number of sentences per chunk and number of overlapping sentences with the previous chunk. The default configuration is 12 sentences per chunk with 2 overlapping sentences, which will be discussed further in the results and discussion section.

The dataset used for development consists of mock interviews between a social worker at SOS and a newly enrolled participant undergoing an initial assessment.

### Survey data

Users can upload an Excel file in a designated format that includes the following fields: question ID, question text, question type, thematic field, and answer options (see Appendix A for an example).

Once uploaded, the entire survey content is wrapped into a single string and added to the prompt context for the language model. At the same time, a Pandas DataFrame is created from the Excel file to serve as a central data structure that stores both the survey definitions and the answers generated in subsequent steps.

As mentioned in the theory section, a semantic search mechanism using vector embeddings was prototyped to improve alignment between survey questions and transcript segments (see [GitHub branch feature/embedding-functions](https://github.com/AmyFeng0227/AI-assisted-survey-tool/tree/feature/embedding-functions)). This involved generating high-dimensional vectors for each survey question and for segments of the transcript (e.g., paragraphs), and then using cosine similarity to match questions to relevant conversation parts. However, this method failed to meet the required accuracy thresholds for reliable MVP performance—some highly relevant survey question matches were consistently missed. Given that modern LLMs support large context windows capable of handling the full survey prompt directly, this semantic matching feature was ultimately deprecated in the current version.

## Pipeline and key features

A diagram of a process

AI-generated content may be incorrect.This section describes the full pipeline following data pre-processing and highlights key features of the MVP. (See *Figure 1* below)

Figure MVP system flowchart

### LLM Prompt Design and Output Handling

To guide the language model effectively, the prompt includes three main components:

1. a text string that enumerates each survey question (with its ID, field category, question text, and answer options)
2. the transcript of the current chunk
3. the previous answers from AI in JSON format (if available)

Prompt engineering techniques outlined in the theory section were applied to optimize accuracy. For example, the prompt includes explicit instructions for output format: for each clearly addressed question, the model must return a JSON object containing the question ID, answer, certainty score, and brief reasoning. A sample output JSON snippet is also included to improve formatting consistency.

The model is instructed not to answer questions that are not explicitly addressed by the participant, and to avoid hallucinations beyond the transcript. This design ensures responses are grounded in actual input and minimizes unnecessary output tokens, thereby reducing latency and cost.

Once the LLM returns a JSON array, it is parsed to extract answers and certainty scores. The system is built to detect invalid JSON formatting—if parsing fails, the prompt is automatically retried up to three times. The retry count is also recorded as part of the evaluation metrics.

### Idempotent Update & data storage

A central feature of the pipeline is its ability to incrementally integrate multiple transcripts without duplicating or overwriting previously captured information. This is accomplished through an idempotent update mechanism, which ensures only genuinely new or updated data modifies the survey state.

Before prompting the language model, the system checks for existing answers stored in answers.json and includes them in the prompt. The model is then instructed to:

* Add answers for previously unanswered questions
* Revise answers if new, conflicting information is provided
* Expand existing answers with new supporting details
* Omit unchanged answers from the new output

This ensures consistency across sessions, reduces redundant processing, and avoids unnecessary token usage.

Updated answers are written back to the central DataFrame created during survey upload. This DataFrame serves as:

* A live record of both survey and answer data
* The data source for the frontend UI
* The basis for the downloadable survey file

Together, these features support a robust, scalable, and user-friendly AI-assisted survey tool aligned with real-world needs in social work settings.

### Transparency and certainty flagging

To enhance transparency and support practitioner oversight, the system includes both a certainty score and a reasoning explanation for each AI-generated answer. For every survey question, the LLM is prompted to output an answer, a certainty level ("low," "medium," or "high"), and a brief rationale summarizing relevant evidence from the transcript. For multiple-choice and single-choice questions, the rationale provides grounding for the AI’s decision, while for free-text questions, where the answer itself serves as a summary, the rationale is left blank. These certainty levels are visually represented in the UI using a color-coded scheme—green for high, yellow for medium, and red for low confidence. This visualization supports human review and offers a helpful starting point for social workers, who always retain final decision-making authority.

### Human-in-the-loop & overrides

Given the sensitive nature of social work, the system is explicitly designed as an assistive tool, not a replacement for human judgment. A human-in-the-loop feature enables users to review and edit answers. In the interface, each question includes an “Edit Answer” button (as an expandable section), allowing users to manually modify both text and multiple-choice responses. Once edited, the certainty level is automatically set to "high", the question becomes locked to prevent further AI modifications, and the entry is highlighted in a darker green to denote human validation.

## Evaluation

This project provides a framework for evaluating trade-offs between accuracy, latency, cost, and user experience by allowing adjustment of chunk size and chunk overlap.

The evaluation used a mock recording (approximately 12 minutes) that was pre-transcribed using Whisper. The accompanying survey included 25 questions, of which 20 were covered in the recording transcript. Three of the covered questions were free-text and excluded from accuracy evaluation due to their subjective nature. Thus, the latency evaluation included all 25 questions, with 22 used for quantitative accuracy metrics.

The test included five chunk sizes (4, 8, 12, 16, 20 sentences) and three overlap settings (0, 2, 4 sentences), resulting in 14 combinations (excluding 4-sentence chunks with 4-sentence overlap). Each configuration was executed three times on the same survey and transcript.

Results were averaged across trials and stored in a CSV file for analysis.

# Result and Discussion

## Evaluation result

A graph with lines and numbers

AI-generated content may be incorrect.

Figure 2 Accuracy, latency, and token cost across different chunking configurations.

The average results from three evaluation runs are visualized above (See Figure 1). Several key trends can be observed: the **accuracy** (blue line) fluctuates between 65% and 80% across different chunk sizes and overlap settings, with no clear linear trend tied to chunk size alone. However, configurations using a **2-sentence overlap** consistently yield slightly higher accuracy than those with 0 or 4 sentences of overlap. This suggests that 2-sentence overlaps may strike a balance by providing just enough contextual continuity between chunks without introducing excessive redundancy that could dilute the model's focus. These findings support maintaining a 2-sentence overlap as the default configuration. Since the product incorporates a human-in-the-loop mechanism, the system can tolerate a certain degree of false positives and false negatives. Additionally, the prompt design in this MVP is intentionally minimal and could be further optimized with added domain-specific knowledge and guidance, which may improve accuracy in future iterations.

The **latency** (yellow line) and **token usage** (green line) indicators show a very strong positive correlation (Pearson r = 0.97). This can be attributed to the fact that the underlying transcript length remains constant, so reducing chunk size increases the total number of chunks. As discussed in the theory section, both latency and token cost are mainly driven by the number of output tokens, which scale with the number of LLM calls—each introducing a relatively fixed processing overhead. Smaller chunks, especially when combined with higher overlap, dramatically increase both token usage and latency. The lowest latency and token costs occur with large chunk sizes and no overlap, while the most resource-intensive configuration involves small chunks with high overlap. Although both metrics behave similarly in this study, they may differ more substantially across different models, making them useful indicators for future tuning.

The **number of retries** is another important factor affecting latency. Each retry adds one or more rounds of processing, increasing both time and cost. In this experiment (see Figure 2), both very large and very small chunk sizes showed slightly higher retry rates. Larger chunks may dilute important context and overwhelm the model, while smaller chunks dramatically increase the total number of requests (from 7 to 64 in this experiment), increasing the chance of formatting or parsing errors overall. Therefore, token cost and latency should be considered alongside retry frequency when deciding on chunking strategy.

A graph with blue and orange lines

AI-generated content may be incorrect.

Figure 3 Total latency and retry frequency across chunking configurations.

However, choosing the configuration with the lowest latency and token cost is not always optimal. This tool is designed to simulate a near-real-time experience. Even with a lower total runtime, larger chunk sizes can lead to longer delays between updates, resulting in a laggy user experience. Smaller chunks allow the user to see more frequent progress, giving the impression of faster response even if the total time is longer. Since the current MVP does not yet support live transcription, this dynamic remains untested but will be crucial in future iterations. For now, a smaller chunk size (8–12 sentences) with 2-sentence overlap is recommended for the current MVP, balancing efficiency, perceived responsiveness, and reasonable token cost.

## Future outlook

Due to time and resource constraints, several future enhancements are proposed for the continued development of this MVP.

### Real-time transcription and interaction

A natural next step would be to evolve from pre-recorded audio processing to real-time listening and transcription. Leveraging models like OpenAI's Whisper with streaming support would enable real-time data capture and feedback, significantly improving usability during live interactions.

### Agentic AI with Domain Knowledge

The introduction of an agentic AI, similar to the techniques described in recent research (e.g., Park et al., 2024), represents a promising direction. This approach would allow the AI to access diverse domain-specific knowledge bases, autonomously retrieving and integrating relevant information, thereby greatly improving analysis depth, reasoning, and accuracy in survey filling and subsequent recommendations.

This functionality will unlock huge potential for the tool, allowing for a standardised and more scientific analysis of the survey, and the functionality could even be expanded to draft the initial individual development according to the survey result and SOS methodology, saving even more time for social workers, and the risk of missing key results from the survey in the development plan is controlled.

With domain knowledge, the tool can be adapted beyond social work into areas such as medical care, mental health, and customer relations management. The underlying methodology of conversation-driven AI-supported data collection, analysis, and automated form completion has broad applicability.

### Technology Accessibility

To improve accessibility in areas with limited connectivity or devices, future iterations could support offline AI features. For example, social workers using paper-based surveys in the field could upload photos of completed forms, which would be parsed and processed by the AI. This would accelerate data processing and make the solution viable in a wider range of operational contexts.

### Sensitive Data Handling and Regulatory Compliance

Future deployments must prioritize data privacy, ethics, and regulatory compliance, particularly under frameworks such as GDPR and the EU AI Act. Given the sensitive nature of social work and the extensive use of personal data, the system may fall under “high-risk” classification, requiring comprehensive documentation, risk management, and governance procedures. Even if the system is not formally classified as high-risk, applying risk assessment and documentation practices will be essential to ensure trust, safety, and ethical deployment.

# Conclusion

This project sets out to explore whether an AI-assisted tool could streamline the process of completing structured surveys in social work settings, particularly during case management interviews. The goal was to maintain high data quality and preserve human oversight, while reducing the manual workload and conversational interruptions that often characterize traditional survey methods. The resulting MVP demonstrates the feasibility of such a solution, combining speech-to-text transcription, large language models, and a user-friendly interface to create a semi-automated workflow that supports—but does not replace—social workers.

The evaluation experiments show that with thoughtful prompt design and chunking strategies, large language models can extract meaningful, structured answers from conversational transcripts with up to 80% accuracy. This is especially promising given that even modest accuracy can deliver value in real-world workflows, where humans will remain in the loop for final review. The trade-off between accuracy and latency was explored in depth, and the findings confirm that careful configuration—particularly around chunk size and overlap—can significantly influence both performance and usability. The best-performing configurations offer a compromise between live-like responsiveness and reliable answer quality.

Key features of the prototype include its transparent decision-making (through reasoning and certainty scores), idempotent data updating logic, and editability features that lock in human input. Together, these create a foundation for trustworthy human-AI collaboration. Importantly, this approach reinforces the value of professional judgment in social work while allowing AI to act as a supportive assistant.

However, this project also highlighted important limitations and areas for future work. Real-time transcription, integration of domain-specific knowledge, and compliance with emerging regulatory frameworks such as the EU AI Act are all critical next steps for scaling this tool in practice. Moreover, exploring accessibility in low-connectivity settings—such as enabling offline processing of paper-based forms—could greatly expand the system's relevance across diverse operational contexts.

In conclusion, this MVP demonstrates that AI can play a valuable role in easing the administrative burden of social workers, enabling them to spend more time on meaningful engagement with participants. While challenges remain, the prototype developed here provides a strong foundation for continued innovation, offering a pathway to more efficient, responsive, and person-centered social services.

# Appendix A example prompt

"""The following transcript is an interview between a social worker and a youth participant interested in participating in the leaving care program. You are provided with the survey (see SURVEY QUESTIONS) which have been partially answered before (see PREVIOUS ANSWERS) based on another transcript. You will update the answers to the survey based on the provided transcripts.

Here is the structure to answer a question:

1. Answer: Base the answer according to the guidance provided in the parentheses. For text questions, try to cover all the relevant information for this question.

2. Certainty (low, medium, high)

3. Text field: All single/multiple choice questions must have a concise text reasoning, but make sure you cover all the relevant information related to the question. If not choice-based, leave blank.

First, you need to recheck the previous answers against the new transcript to detect any potential conflicts or new information.

- If the new transcript contains conflicting information, update the previous answer according to the current transcript.

- If the new transcript contains additional/new information, update the previous answer by adding the new information while keeping the previous answer.

- If the new answer is similar to the previous answer, no need to update.

Second, find answers in the new transcript for questions not answered previously:

- Only fill out the answer if the transcript has clearly addressed the question.

important:

- Answers to single or multiple-choice questions should be in a list.

- Only answer the questions that are clearly addressed in the transcript.

- Output ONLY for the updated answers and newly answered questions.

- Do not make up information, follow the transcript.

- Format your response as a JSON array, nothing else.

SURVEY QUESTIONS:

{*survey\_questions*}

PREVIOUS ANSWERS:

{*previous\_answers*}

NEW TRANSCRIPT:

{*transcript*}

output example:

[

  {{

    "question\_id": "5",

    "answer": ["housing", "education"],

    "certainty": "high",

    "text field": "support in finding an apartment is urgent. Prefer first-hand contract"

  }},

  {{

    "question\_id": "10",

    "answer": "lonely and depressed, having trouble to sleep and hard to find time for friends",

    "certainty": "medium",

    "text field": ""

  }}

]

"""

# Appendix B example survey



Bibliography

IBM. (n.d.). *What is an attention mechanism?* IBM. Retrieved July 21, 2025, from <https://www.ibm.com/think/topics/attention-mechanism>

Kilpatrick, L., Mallick, S. B., & Kofman, R. (2024, June 27). *Gemini 1.5 Pro 2 M context window, code execution capabilities, and Gemma 2 are available today*. Google Developers Blog. https://developers.googleblog.com/en/new%20features%20for%20the%20gemini%20api%20and%20google%20ai%20studio/

OpenAI. (2023, March 27). *GPT-4 technical report*. <https://openai.com/index/gpt-4-research/>

OpenAI. (n.d.). *Compare models*. OpenAI Platform. Retrieved July 22, 2025, from <https://platform.openai.com/docs/models/compare>

OpenAI. (n.d.). *Latency optimization*. OpenAI Platform. Retrieved July 23, 2025, from <https://platform.openai.com/docs/guides/latency-optimization>

OpenAI. (n.d.). *Optimizing LLM accuracy (Guide)*. OpenAI Platform. Retrieved July 23, 2025, from <https://platform.openai.com/docs/guides/optimizing-llm-accuracy>

OpenAI. (n.d.). *Reasoning models (Guide)*. OpenAI Platform. Retrieved July 22, 2025, from <https://platform.openai.com/docs/guides/reasoning?api%20mode=chat>

Park, J. S., Zou, C. Q., Shaw, A., Hill, B. M., Cai, C., Morris, M. R., ... & Bernstein, M. S. (2024). *Generative agent simulations of 1,000 people*. arXiv preprint arXiv:2411.10109.