SOCbot: Using Large Language Models to dynamically measure and classify occupations in surveys

Patrick Sturgis[1] Thomas S. Robinson[1] Laura Fung[1] Caroline Roberts[2]

[1]Department of Methodology, London School of Economics and Political Science

[2]Institute of Social Sciences, University of Lausanne

Survey Futures Workshop: Industry and Occupation Coding
UCL, 4 June 2026

LSE Department of Methodology

SURVEY FUTURES
SURVEY DATA COLLECTION METHODS COLLABORATION

What SOCbot does

An LLM embedded in the questionnaire scripting software that codes occupation in real time, and probes for more detail when it is unsure.

Two components

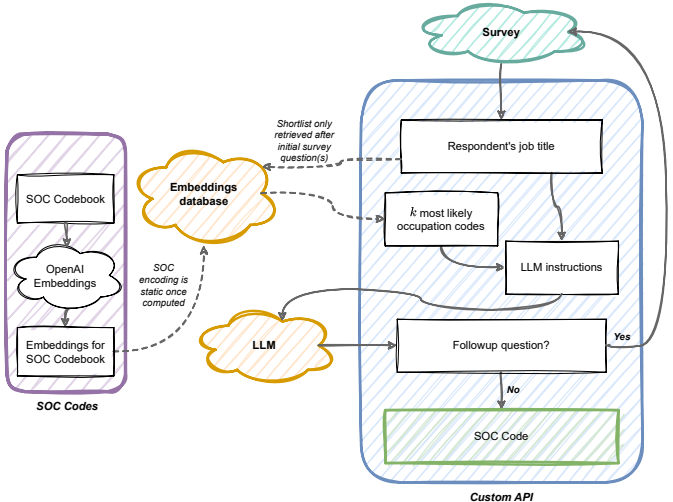
1. A **classifier** (retrieval-augmented) that maps a response to a SOC unit group.
2. **Dynamic probing**: tailored follow-up questions generated when coding confidence is low.

Two modes

- **Offline**: re-code archived survey responses with any model.
- **Live**: classify and probe inside a running survey.

Key innovation: the LLM replicates interviewer-style probing, adapting each follow-up to the gaps in the respondent's own answer.

The pipeline



Dynamic follow-ups

When confidence is low, SOCbot returns a question instead of a code; the answer is fed back and it tries again.

- Probes are constrained to job-relevant content: industry/sector, job tasks, qualifications, and supervisory responsibility.
- It can also re-ask when a title is garbled (e.g. "Accnt"), re-triggering retrieval on the cleaned title.
- The set of probes is extensible (hours, pay basis, and so on).
- A cap on the number of probes lets the researcher trade accuracy against respondent burden.

Live deployments use fast non-reasoning models for latency; responses can be re-classified offline with a reasoning model later.

Study 1: how well does the static classifier code?

Benchmark SOCbot against the existing production-coded comparator (CASCOT + human adjudication) on two UK surveys.

- **Public Voice** probability panel ($n = 2{,}000$): job title and industry.
- **Next Steps** longitudinal study ($n = 5{,}162$): title, tasks, industry, qualifications.

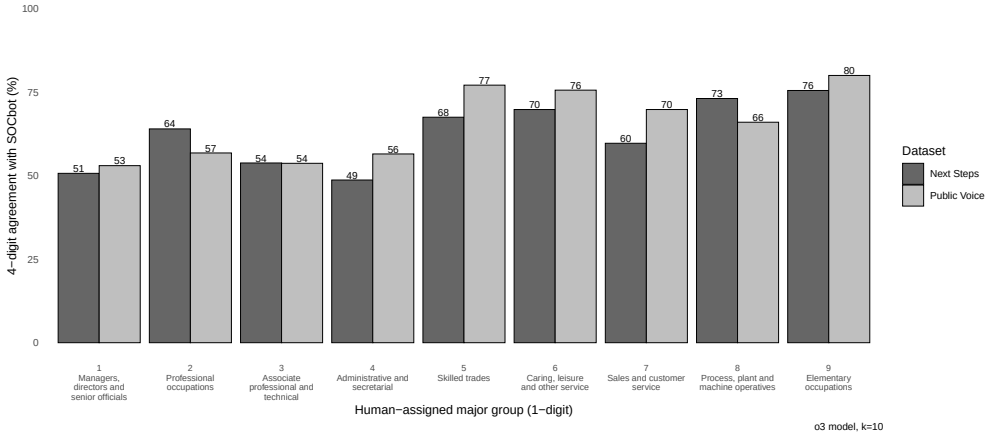
No true value exists, so we report inter-coder agreement and benchmark it against the published human-only literature (0.4 to 0.7 at the unit-group level).

Agreement is comparable to human coders

Public Voice (n=2000), k=10	Major	Submajor	Minor	Unit
`o4-mini` (original)	0.797	0.754	0.709	0.599
`o3` (original)	0.826	0.786	0.742	**0.624**
`Opus 4.6`, no thinking	0.809	0.768	0.726	0.614
`Opus 4.6`, adaptive reason	0.800	0.762	0.721	0.614
`GPT-5`	0.811	0.769	0.728	0.608
`gpt-oss-120b` (open)	0.778	0.733	0.675	0.558
`gpt-oss-20b` (open)	0.770	0.724	0.671	0.547

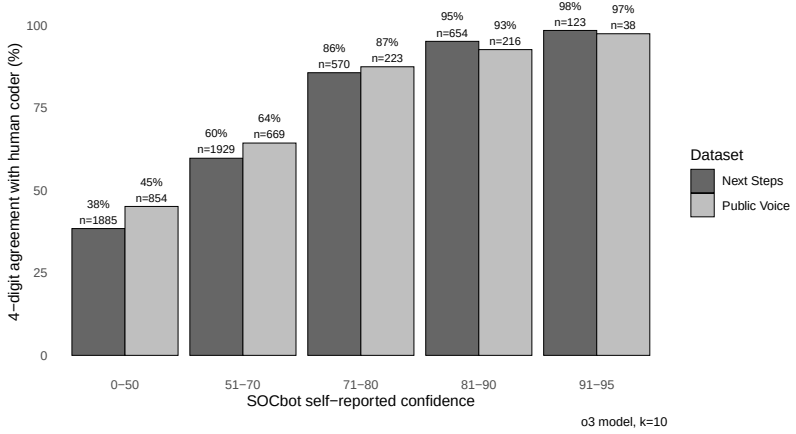
Organisations that cannot send personal data to a commercial API (statistics agencies, health surveys) can still run SOCbot, at a modest and measured accuracy cost.

Where SOCbot and human coders disagree



Hardest: managerial, administrative, associate-professional. Easiest: skilled trades, caring, elementary. Healthcare is a particular strength.

Confidence is a usable triage signal



Agreement rises monotonically with self-rated confidence: $\geq$71 gives >90%; $\leq$50 gives 38–45%.

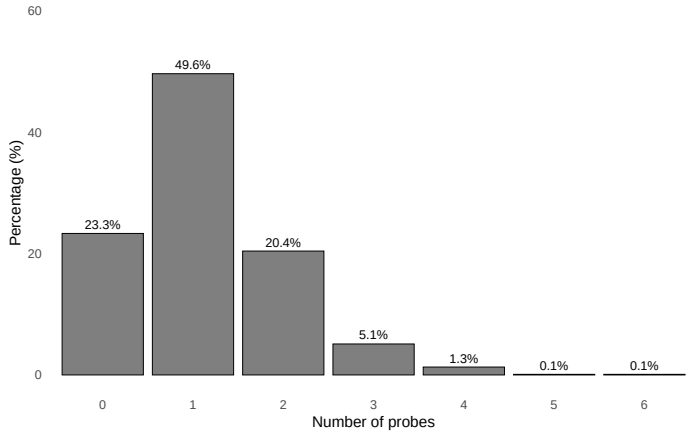
Auto-accept high-confidence cases, route the rest to human review.

Study 2: the full pipeline in a live survey

Verian UK Public Voice panel, fieldwork 7–10 July 2025, Forsta+ scripting.

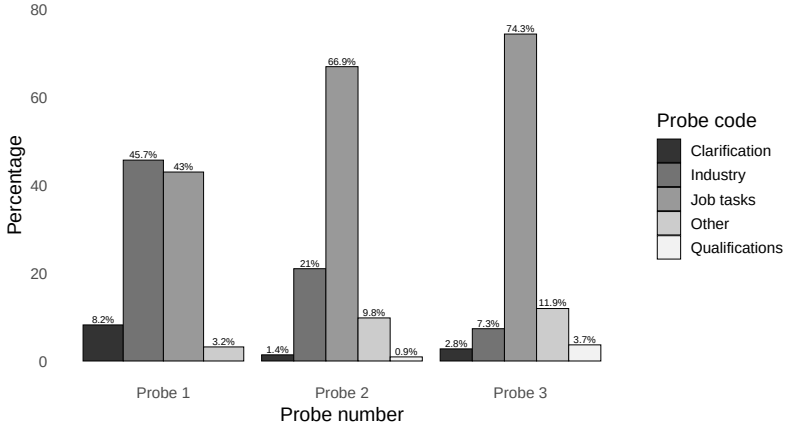
- 8,770 invited; 1,720 started (19.6%); 7% declined LLM consent.
- Panel members had previously given a job title coded to SOC20 by humans, so we can compare before and after probing.
- Analysis restricted to the 1,093 respondents whose job title was unchanged.
- Sample skews graduate (61%) and male (55%).

75% of respondents can be coded after just two probes



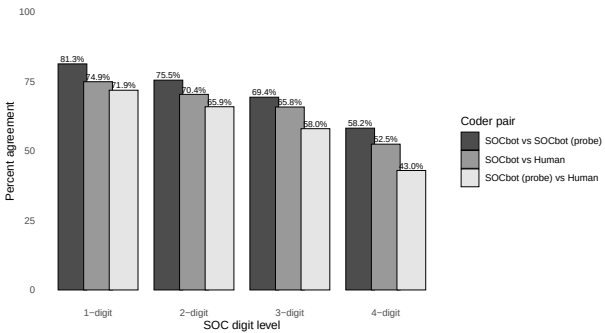
23% coded on job title alone; 50% on title plus one probe; under 2% needed more than three. Total burden is close to the standard two-question approach, but distributed very differently.

Probes are tailored on previous responses



First probe splits between tasks and industry; later probes shift to tasks. 8% of first probes clarify a garbled title, which a static survey cannot do. A fixed two-question script is therefore sub-optimal.

Reliability and the accuracy ordering



Dynamic SOCbot sees the most information, so treat it as the most accurate coding, a benchmark.

Read the bars as closeness to that benchmark. At the unit-group level the static classifier agrees with dynamic SOCbot on 58% of cases, but the human coder agrees with it on only 43%.

Being closer to the best-informed coder, the static classifier is more accurate than the human coder. Inferred ordering:

1. Dynamic SOCbot
2. Static SOCbot classifier
3. Human coder

Is it reproducible, and how does it handle bad input?

Same input, same code?

Five reruns of 200 cases (`Opus 4.6`, no reasoning):

- Identical 4-digit code on all five runs for 84% of cases (94% at 1-digit).
- The cases that vary across runs are almost all low-confidence ones.
- OpenAI's `seed` parameter gave no measurable gain.

For guaranteed reproducibility, self-host an open-weight model with a fixed seed.

Behaviour on malformed input

11 hand-curated stress cases:

- Contradictory title and industry: resolves to the concrete description, at reduced confidence.
- Nonsense input: confidence $\leq$10.
- Prompt injection: ignored.
- Heavy abbreviations: decoded, or flagged when ambiguous.

Takeaways

- SOCbot matches or exceeds human coding accuracy, in both static and dynamic use.
- It removes the separate post-fieldwork coding stage, cutting cost and turnaround.
- Dynamic probing tailors the questions asked, improving accuracy at similar overall burden.
- The confidence score gives a calibrated, tunable triage signal for production pipelines.
- Results are stable across four model families, including on-premise open-weight models.

Thank you

patrick.sturgis@lse.ac.uk

Sturgis, P., Robinson, T., Fung, L. and Roberts, C. (2026) 'SOCbot: Using Large Language Models to measure and classify occupations in surveys'. *Sociological Methods and Research*. In Press.

Prototype codebase: `anonymous.4open.science/r/soc_flask-657D`