WEBINAR

Artificial Intelligence and the Future of Survey Research

Wednesday, July 22

RTI INTERNATIONAL

The Washington Statistical Society WSS SINCE 1926

David Wilson, PhD

RTI International

- Senior Director of Statistics at RTI International with 26 years of experience in survey sampling, statistical weighting, imputation, nonresponse bias analysis, and statistical disclosure control.

- Over twenty years of experience leading statistical activities on national secondary education studies including the Education Longitudinal Study of 2002, the High School Longitudinal Study of 2009, the Middle Grades Longitudinal Study of 2017, High School and Beyond 2022, the International Computer and Information Literacy Study of 2023, and the Trends in International Mathematics and Science Studies of 2023 and 2027.

- He also directs RTI's SUDAAN software, a widely used tool for the analysis of complex survey data, and has developed disclosure control methodology used on a variety of projects including the National Survey on Drug Use and Health.

Dr. Anna-Carolina Haensch

University of Maryland

- Vox Populi, Vox AI: used large language models to estimate German vote choice from GLES survey data, testing whether LLMs can reproduce individual-level political behavior

- Automating Survey Quality Prediction: applying LLMs to automate the manual coding behind the Survey Quality Predictor (SQP), replacing an expensive expert-annotation bottleneck.

- AI in questionnaire design & evaluation: investigating where LLMs can support instrument development and pretesting in survey methodology.

- Zero-shot text-to-code classification: supervising work that maps free-text responses to official statistical classifications (ISIC/NACE) without labelled training data.

Lance Couzens

RTI International

- Thought leader and promoter of knowledge sharing events in the open source and AI spaces
- Developer and maintainer of multiple open-source packages, and an early adopter of AI to support that work
- Founder and leader of RTI's Open Source Mentors team
- Leader of RTI's Claude Code Pilot, focusing on agentic workflows to support statistical practice

Building National School Sampling Frames with LLMs

David Wilson

RTI International

Outline

- Goal
- Sampling Frame Construction
- Single and dual LLM approaches
- Development with LLMs
- Results
- Lessons Learned

Goal

- Determine viability of using LLMs to be able to construct school sampling frames.
- By comparing human-prepared school sampling frames constructed for TIMSS 2027 to LLM-generated sampling frames.

TIMSS 2027 School Sampling Frames

- Two school sampling frames: public- and private- schools in the 50 United States and District of Columbia offering instruction in grade 4 and grade 8.

- Frames include 34 variables for each school:
  - Stratification variables
  - Sorting variables
  - ID and geographic variables including mailing and physical addresses
  - School characteristic flags (virtual schools, special education schools, schools embedded in hospitals, schools embedded in treatment facilities, other residential schools)
  - District and Diocesan identifiers

Single and dual LLM approaches

- Two approaches; single LLM and dual LLM
- Single LLM: Claude Sonnet 4.6
- Dual LLM: Chat GPT 5.4 mini for construction, Claude Sonnet 4.6 for validation
- Scan input files, create R code to construct and validate constructed sampling frames

Development

- Copilot CLI
- Provided CCD (Common Core of Data, NCES, SY 2024–25) files: public-school directory, lunch-program eligibility, membership counts, school characteristics, and EDGE geocode files
- Provided PSS (Private School Survey, NCES, 2021–22) data file
- Companion documentation: file layouts, codebooks, data notes, and release notes

Development (cont’d)

- Iterated with LLM to figure out how to provide information to the LLM so that it produced quality sampling frames.
- Created a new input file; an excel file, variable_dictionary.xlsx, containing a description of the frame's 34 output variables.
- Included variable definitions, construction rules, and missing data rules.
- Added two more input files, CSV files for the LLM to use to map Dicoese codes to names and to map FIPST codes to State and Region.

Development (cont’d)

- Created one instruction file with frame construction information for LLM.
- Created a second instruction file for validation of the constructed frames.

Results

- The LLM-produced frames were compared cell-by-cell against frames built by a methodologist using conventional R code. The comparison covered 3,797,086 individual data values (111,679 records × 34 variables).
- 189 differences (>99.995% agreement) and the LLM value was correct 136 of the 189 times (72%) where there were differences.
- 177 of the differences occurred with flags identifying schools embedded in hospitals or correctional facilities.
- The LLM did a better job of identifying such schools by using a broader set of keywords when scanning school names; though it did produce some false positives.

Lessons Learned

- Create file containing variables, variable definitions, construction rules, and as much information (file source) as feasible.
- The hardest variable for the LLM to construct was poverty ratio; which is a function of three counts (from two different source files.)
- The dual-LLM approach was able to replicate the same output as the single-LLM approach and takes less tokens than the single-LLM approach.
- The dual-LLM approach required much more explicit guidance around construction of poverty ratio than the single-LLM.

Lessons Learned (cont'd)

- The dual LLM approach misused the R coalesce() function. Had to modify instructions so that it did not attempt to use the coalesce() function.
- The LLM-produced sampling frames are comparable in quality to human-constructed sampling frames.
- The users of LLMs must still have knowledge about the CCD and PSS data files and how variables should be constructed or derived from variables in those sources.

Thank you

Contact: David Wilson

Email: dwilson@rti.org

Munich Center for Machine Learning

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Using LLMs for Coding German Open-Ended Survey Responses on Survey Motivation

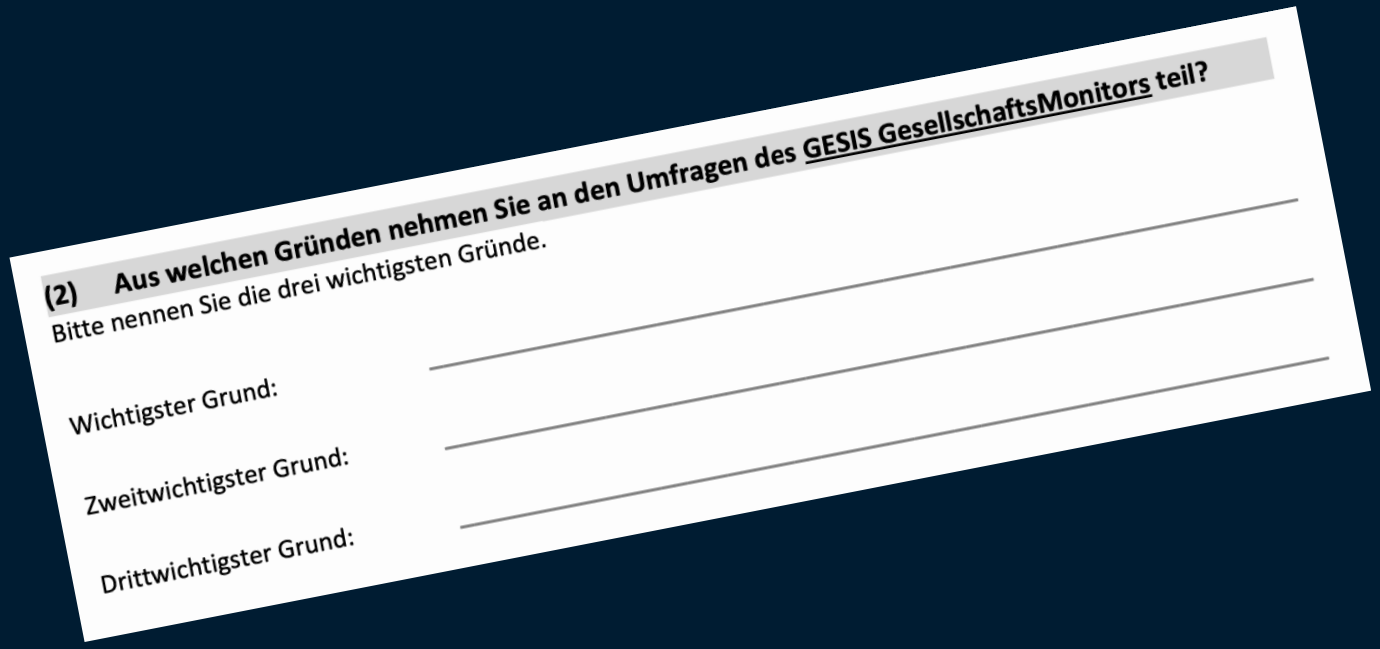
Anna-Carolina Haensch
LMU Munich | Munich Center for Machine Learning | University of Maryland College Park

*work with **Leah von der Heyde**, Bernd Weiß, Jessica Daikeler*

Munich Center for Machine Learning

Research Question

Can LLMs be used to code open-ended survey responses from surveys? How much context is needed?



(2) Aus welchen Gründen nehmen Sie an den Umfragen des GESIS GesellschaftsMonitors teil?

Bitte nennen Sie die drei wichtigsten Gründe.

Wichtigster Grund: _____

Zweitwichtigster Grund: _____

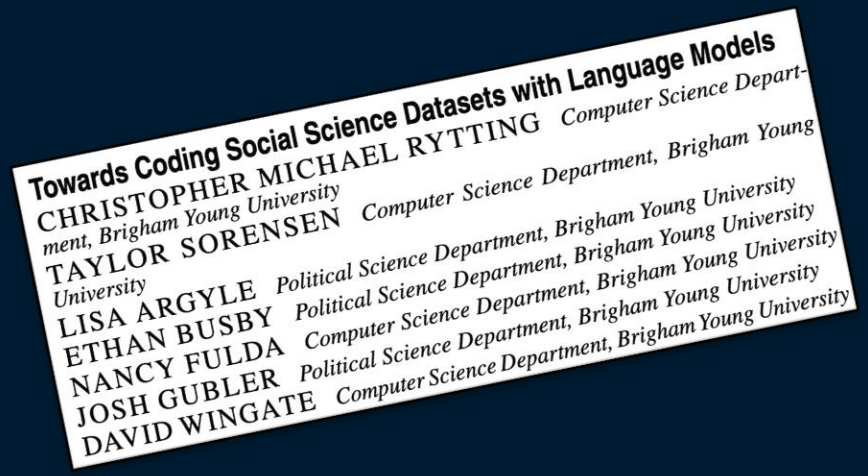
Drittwichtigster Grund: _____

Research Question

Munich Center for Machine Learning

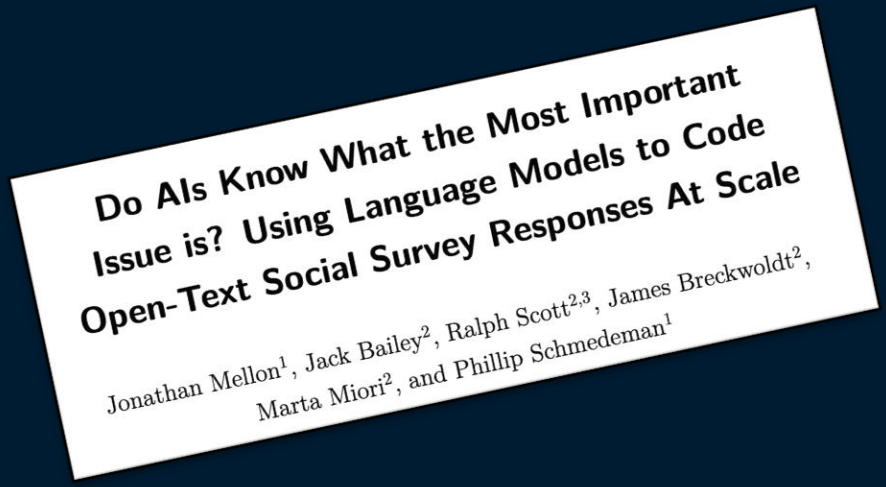
Can LLMs be used to code open-ended survey responses from surveys? How much context is needed?

Lots of interest in this question, some examples for previous research:



Towards Coding Social Science Datasets with Language Models

CHRISTOPHER MICHAEL RYTTING *Computer Science Department, Brigham Young University*
TAYLOR SORENSEN *Computer Science Department, Brigham Young University*
LISA ARGYLE *Political Science Department, Brigham Young University*
ETHAN BUSBY *Political Science Department, Brigham Young University*
NANCY FULDA *Computer Science Department, Brigham Young University*
JOSH GUBLER *Political Science Department, Brigham Young University*
DAVID WINGATE *Computer Science Department, Brigham Young University*



Do AIs Know What the Most Important Issue is? Using Language Models to Code Open-Text Social Survey Responses At Scale

Jonathan Mellon[1], Jack Bailey[2], Ralph Scott[2,3], James Breckwoldt[2], Marta Miori[2], and Phillip Schmedeman[1]

Munich Center for Machine Learning

Research Question

Can LLMs be used to code open-ended survey responses from surveys? How much context is needed?

Our case study:

- Non-English language: German
- Complex problem (around 20 categories in the classification scheme)
- Comparing several levels of provided context as well as three popular LLMs

mcml Data
Munich Center for Machine Learning

(2) Aus welchen Gründen nehmen Sie an den Umfragen des GESIS GesellschaftsMonitors teil?
Bitte nennen Sie die drei wichtigsten Gründe.

Wichtigster Grund: _____

Zweitwichtigster Grund: _____

Drittwichtigster Grund: _____

GESIS Panel

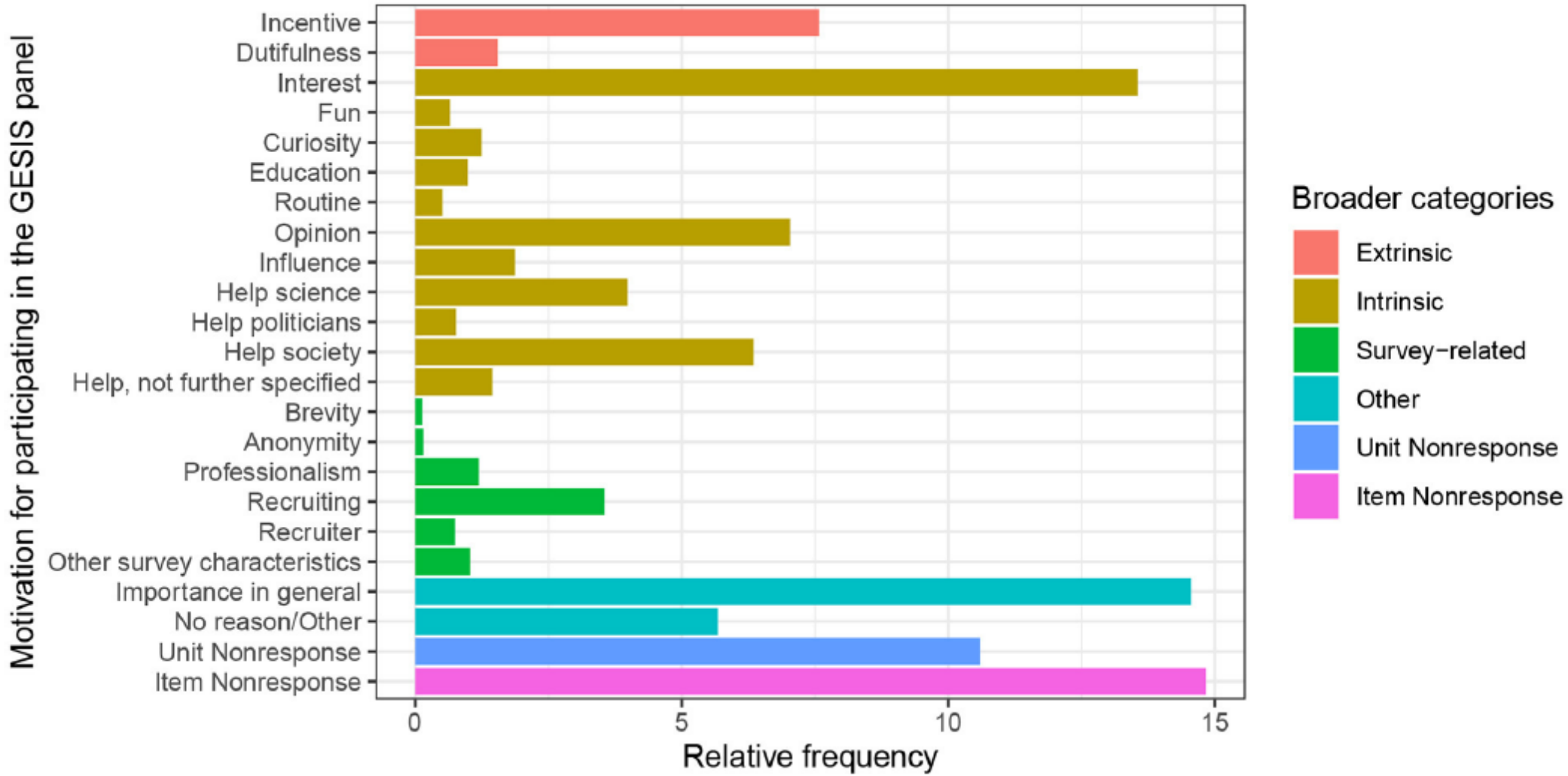
Translation:

(2) For what reasons do you participate in the surveys of the GESIS GesellschaftsMonitor?
Please name the three most important reasons.

Most important reason: _____
Second most important reason: _____
Third most important reason: _____

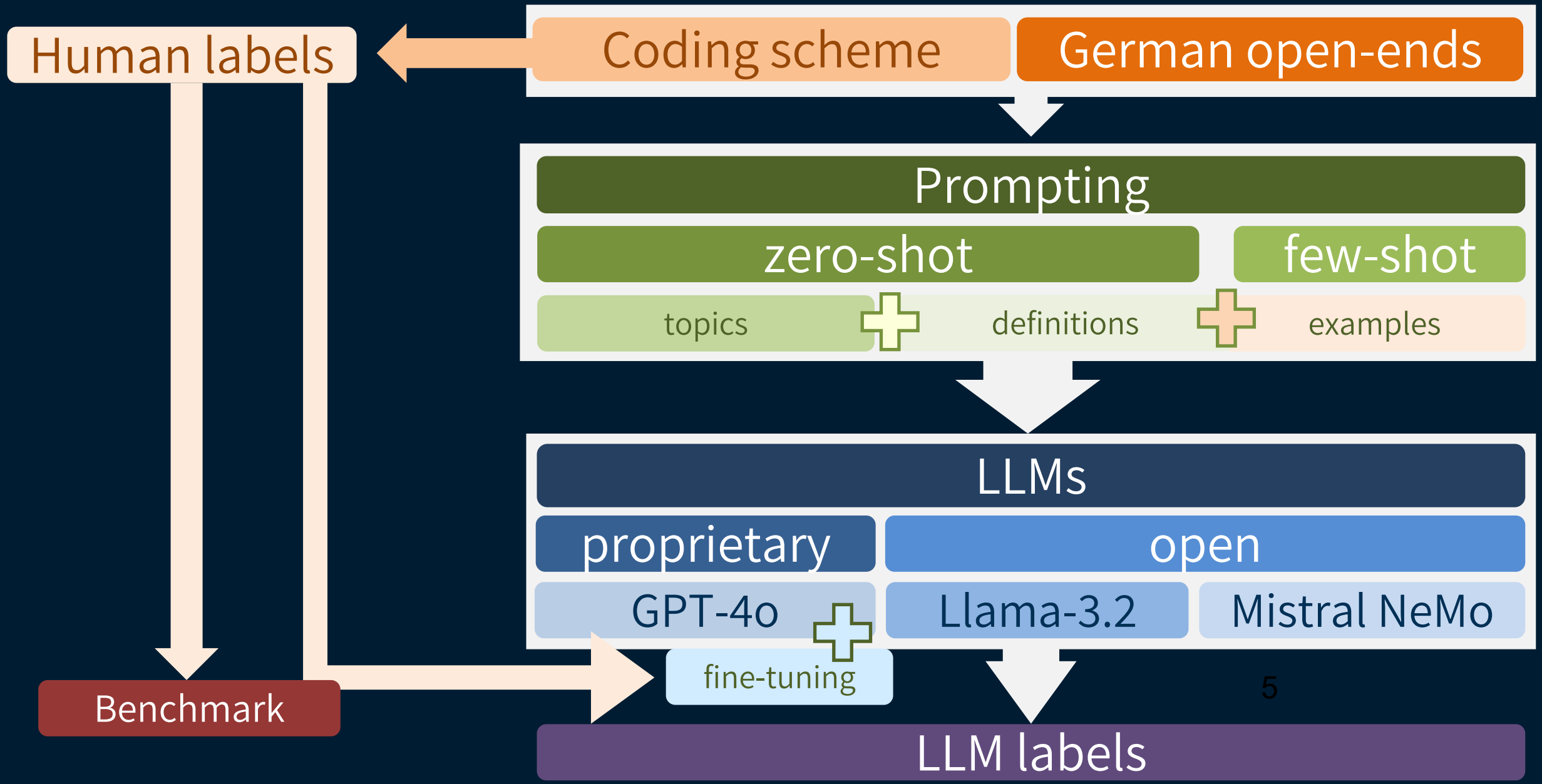
Munich Center for Machine Learning

Human coding scheme



Experimental Set-Up

Munich Center for Machine Learning



Munich Center for Machine Learning

Experimental Variation

You are a survey expert classifying open-ended responses to the question why individuals participate in a survey. Assign these reasons for participating to exactly one of the following categories.
The categories are:

INTEREST: Description
TELL OPINION: Description
INFLUENCE: Description
INCENTIVE: Description
FUN: Description
ROUTINE: Description
….
IMPORTANCE IN GENERAL: Description
OTHER: Description
NO REASON: Description

Variation 1: Add descriptions

Make your best guess, even if it is hard.
Respond in the following format: Reason for participating | CATEGORY.
Do not give an explanation for your classification, but return only the reason for participating and your classification.

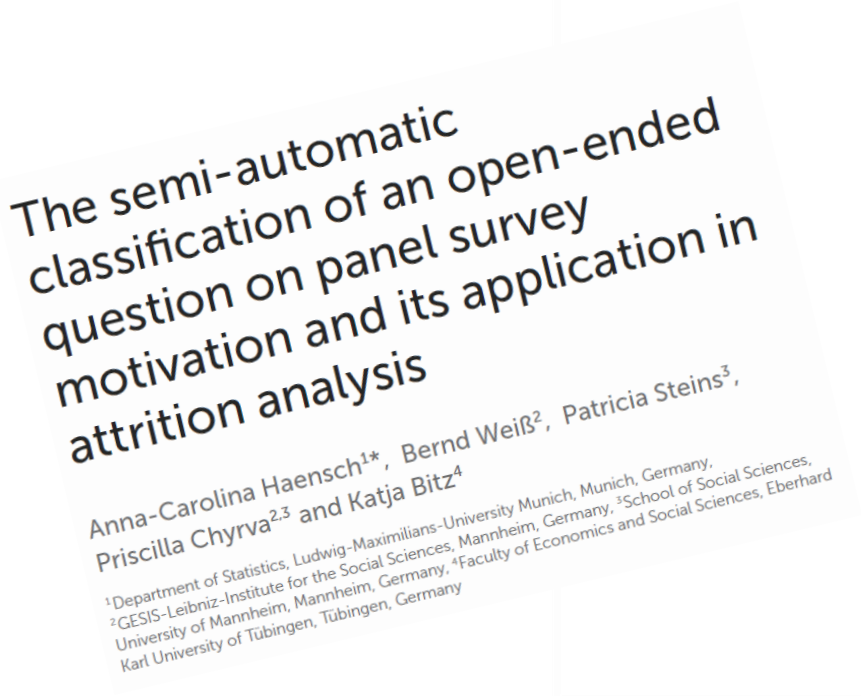
Examples:
[Example reason | CATEGORY 1] …
Classify the following reason for participating:
[open-ended response]

Variation 2: Add examples (few-shot prompting)

Munich Center for Machine Learning

Results

Results


The semi-automatic classification of an open-ended question on panel survey motivation and its application in attrition analysis

Anna-Carolina Haensch[1*], Bernd Weiß[2], Patricia Steins[3], Priscilla Chyrva[2,3] and Katja Bitz[4]

[1]Department of Statistics, Ludwig-Maximilians-University Munich, Munich, Germany, [2]GESIS-Leibniz-Institute for the Social Sciences, Mannheim, Germany, [3]School of Social Sciences, University of Mannheim, Mannheim, Germany, [4]Faculty of Economics and Social Sciences, Eberhard Karl University of Tübingen, Tübingen, Germany

Summary

Munich Center for Machine Learning

Can LLMs be used to code open-ended survey responses from surveys? How much context is needed?

- State-of-the-art LLMs do not always outperform much simpler bag-of-words methods like SVM or logistic regressions
- For a complex problem, fine-tuning an LLM was needed to achieve satisfactory results.
- Context is key and helpful

Munich Center for Machine Learning

Questions? Collaborations? Let's connect!

Full Paper (https://ojs.ub.uni-konstanz.de/srm/article/view/8568)

Caro Haensch
LMU Munich | Munich Center for Machine Learning | University of Maryland College Park

C.Haensch@lmu.de

AI & THE FUTURE OF SURVEY RESEARCH · WASHINGTON STATISTICAL SOCIETY

Eliminating the First-Draft Burden

Lance Couzens

RTI International

WHY DOCUMENTATION IS THE BOTTLENECK

Documenting a complex statistical product is critical to its use — and it devours time the team would rather spend on the method.

Critical

Without clear methodology, a product can't be reviewed, trusted, defended, or reused.

Slow

The blank page is the tax. Synthesizing scattered context into a first draft eats weeks.

The shift

Let an agent synthesize the drafts *and* carry much of the revision load — while you steer through normal Word review.

THE WORKFLOW AT A GLANCE

A round-trip between Markdown and Word

01 AGENT

Author

Draft in Markdown; render to Word.

md → docx

→

02 HUMAN

Review

Reviewers mark up in Word — track changes & comments.

.docx

→

03 AGENT

Ingest

Extract edits + comments back to Markdown.

docx → md

→

04 AGENT

Revise

Agent edits the source under your direction.

md*

→

05 AGENT

Redline

Clean docx + a Word-native "what changed."

md → docx ±

↻ REPEAT

The same loop runs through **internal reviewers, then external delivery** — each pass tightens the document. The Markdown source stays the single source of truth throughout.

The draft is only as good as the workspace

Populate the workspace

Scripts/	the production code — the ground truth for what the method actually does
Reference/	early methodology notes, prior memos, related write-ups
StyleGuide/	house conventions, terminology, formatting expectations
Prompts/	the structured prompt that frames the task

Why it works

- The agent **synthesizes from real context** — code and references — not a blank page.
- Grounding the draft in the actual scripts keeps the methodology **faithful to implementation**.
- You move from **author to editor** — shaping and correcting instead of synthesizing.

THE STRUCTURED PROMPT

Frame the task, don't just ask for it

<instructions> Draft a technical memorandum documenting a bespoke nowcasting methodology for a high-profile monthly statistic . Leverage the early methodology in Reference/ and the final process as implemented in Scripts/. **</instructions> <role>** You are a team of senior research statisticians documenting the method for a technical audience that will scrutinize every assumption. **</role> <context>** The project team is supportive but must satisfy an adversarial reviewer. Timeline and realistic data constraints are relevant. **</context> <format>** Author in Markdown; convert to .docx via pandoc. **</format>**

The tags aren't required — they **organize the request** for you as much as the agent: what to do, who's writing, what pressures shape it, and how to deliver.

Reviewers stay in Word. The source stays in Markdown.

3 **Ingest with fidelity**

pandoc reads the marked-up docx with **--track-changes=all** — the flag that preserves insertions, deletions, *and* comments (the default silently drops them).

4 **Revise the source**

Apply edits literally; treat comments as intent; harmonize parallel passages. Any change the reviewer didn't explicitly mark is flagged for you to accept or veto.

5 **Mechanical redline**

docx-compare drives Word to diff clean vs. revised, producing native tracked changes. "What changed" is *generated*, not hand-assembled.

HOW A REVIEW READS ONCE INGESTED

On this project, QC is ~~a back-office function~~ integral to production. ◆ "Soften this absolute claim — check other instances too."

Encoded as inline spans pandoc round-trips losslessly: [inserted]{.insertion author=…} [deleted]{.deletion author=…} [note]{.comment-start id=…}

Why it works, and where it doesn't

Why it beats a first draft

+ It's **easier to push back and shape** than to synthesize from nothing.
+ Reviewers **keep their tools** — ordinary Word track changes and comments.
+ The Markdown source stays **diffable and version-controlled**.
+ "What changed" is **mechanical and verifiable** , not assembled by hand.

Caveats

! Built for **technical writing** — accuracy over style, flair, or voice.

! **Model choice matters** a great deal; frontier models are markedly better at nuanced prose.

! The redline step needs **Word on Windows** ; every other step is portable.

! You still **own the document** — review is the quality assurance.

TAKEAWAYS

Ship documentation faster — without giving up control

01 The agent **synthesizes the drafts and carries much of the revision load** — not just the first draft; you direct.

02 Reviewers work in **plain Word**; Markdown stays the source of truth.

03 Every change is **captured in a mechanical redline** you can verify.

TRY IT

Install **pandoc** and run this loop with your preferred agentic toolkit — or, as a stakeholder, **ask for this workflow** on your next deliverable.