Working paper ZMB-25058 October 2025

Transforming Zambia's Labour Force Survey using artificial intelligence

Tyler Rossow
Tobias Edison
Rory Hardie
Harry Mayne

IGC
DIRECTED BY LSE UNIVERSITY OF OXFORD
FUNDED BY UKaid from the British people

Transforming Zambia's Labour Force Survey Using Artificial Intelligence

Tyler Rossow[1], **Tobias Edison**[1], **Rory Hardie**[1] **and Harry Mayne**[1,2]
[1]International Growth Centre, [2]University of Oxford

National statistics agencies conduct labour force surveys to understand employment and sectoral trends, providing crucial evidence for industrial, fiscal, and monetary policies. Typically, human enumerators assign numerical occupation and industry codes to survey responses, a time-consuming and error-prone process. In this work, we show that large language models (LLMs) can substantially improve classification accuracy at relatively low cost. In collaboration with the Zambia Statistics Agency, we develop an LLM-based pipeline to classify 1,000 Labour Force Survey responses into occupation and industry codes. All evaluated LLMs outperform human enumerators by a statistically significant ($p < 0.001$) margin, with GPT-5 Nano doing so at the lowest cost. Our approach, which we believe is the most advanced to date in an African context, reshapes the understanding of Zambia's economic composition and can scale to other countries and survey contexts.

Corresponding author: Rory Hardie at roryj.hardie@theigc.org

1. Introduction

Zambia's Labour Force Survey (LFS) is a household-based sample survey administered by the Zambia Statistics Agency (ZamStats) in partnership with the Ministry of Labour and Social Security. The primary objective of the LFS is to determine the size of the labour force and analyse patterns across age, education, gender, occupation, and industry. Since 2017, the quarterly LFS has been the basis for nationally representative statistics on Zambia's labour market, underpinning industrial, fiscal, and monetary policies.

Currently, human enumerators conducting the survey in the field use verbal responses to assign four-digit occupation and industry codes. The codes are assigned based on the International Standard Classification of Occupations (ISCO) and International Standard Industrial Classification (ISIC), a standardised process that is time-consuming and error-prone for human enumerators. In this paper, we introduce an automated method for ISCO and ISIC coding using large language models (LLMs). A retrieval-augmented generation (RAG) pipeline restricts the set of possible code choices; then, with few-shot prompting and the official classification codebooks, we use LLMs to select the most appropriate code.

This method substantially outperforms human classification (Figure 1). On ISCO, the system identifies the correct code in 60-63% of cases (from 436 choices); on ISIC, the system identifies the correct code in 57-66% of cases (from 419 choices). Our results represent an improvement of 10-17 percentage points upon human enumerators and are statistically significant at the 99.9% confidence level. With GPT-5 Nano, we can obtain advanced performance for under $10 per annual LFS sample of 10,400 households.

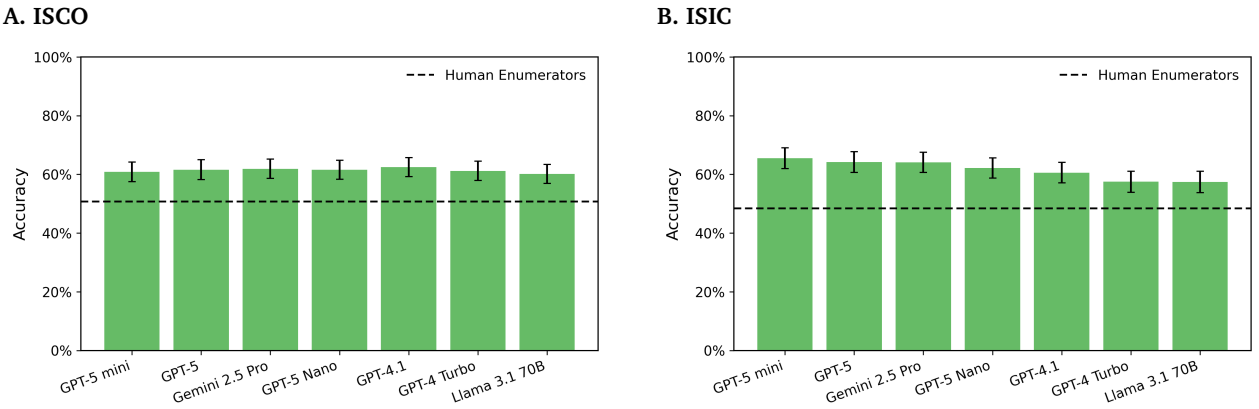


Figure 1: LLMs Outperform Human Enumerators. LLMs are compared against human enumerators on the labelling task assigning four-digit **A.** International Standard Classification of Occupations (ISCO) and **B.** International Standard Industrial Classification (ISIC) codes. Human enumerators and LLMs assign codes based on textual responses to Zambia's Labour Force Survey. All LLMs outperform enumerators with statistical significance at the 99.9% confidence level. Error bars are reported with 95% Wald confidence intervals for the pairwise difference between each LLM and enumerators.

After extensive analysis of LLM and enumerator errors, we believe that this is the maximum performance possible using our current dataset. The remaining errors largely stem from incomplete descriptions or inaccurate ground truth codes.

The contributions of our work are:

- A low-cost, high-accuracy classification method, which is the most advanced to date in Africa.
- A reshaped understanding of Zambia's economic composition, informing industrial policies.
- A faster method to produce official statistics, improving countercyclical fiscal and monetary policy.
- A scalable methodology that can be applied to other countries and surveys.

2. Background

In this section, we provide a background to ISCO and ISIC classification, including current challenges.

2.1. ISCO and ISIC classification

ISCO and ISIC are standardised classification systems maintained by the International Labour Organization (2008) and the United Nations Statistics Division (2008), respectively. ISCO codes classify *jobs* based on the duties performed by the worker, while ISIC codes classify *industries* based on the main economic activities of the establishment in which the worker operates. In both cases, the level of granularity increases with each relevant digit, e.g., the first ISCO digit is a more general description than all four digits together. Table 1 explains each digit and its level of detail for the ISCO-08 and ISIC Rev. 4, the editions used by ZamStats. We also include an example from the LFS dataset for illustration.

Currently, hired enumerators conduct the survey using electronic questionnaires in face-to-face interviews. The core component of the LFS questionnaire is the *Characteristics of the Main Job* module, which seeks to identify an individual's occupation and industry. The questions asked are:

- **Occupation:** "In his/her main job/business, what kind of work does (NAME) usually do? (Write occupation title, if any, and main duties and tasks)"
- **Industry:** "In (NAME) workplace what kind of business/activity is mainly carried out? (Write name of establishment, if any, and main activity, goods, or services)"

As respondents answer these questions, enumerators record information about the respondent's job title, job description, and the main activities of their place of employment. This information is used by enumerators to assign ISCO and ISIC codes. While all information can be relevant for assigning codes, the ISCO code is primarily assigned based on job titles and descriptions, while the ISIC code is primarily assigned based on the main activities of the employing firm.

Number of Digits	ISCO	ISIC
One	Major Group **6**310 – Skilled Agricultural, Forestry, and Fishery Workers	N/A
Two	Sub-major Group **63**10 – Subsistence Farmers, Fishers, Hunters, and Gatherers	Division **01**11 – Crop and Animal Production, Hunting, and Related Service Activities
Three	Minor Group **631**0 – Subsistence Crop Farmers	Group **011**1 – Growing of Non-Perennial Crops
Four	Unit Group **6310** – Subsistence Crop Farmers	Class **0111** – Growing of Cereals (Except Rice), Leguminous Crops, and Oil Seeds

Table 1: ISCO and ISIC Codes Explained. The table illustrates the progressively finer level of detail communicated by ISCO and ISIC digits, using the example of a subsistence crop farmer (grey). Survey records for this respondent contain:

`D1_TITLE`	Subsistence Crop Farmer
`D1_DESC`	Subsistence Crop Farmer Growing of Maize
`D2_MAIN_ACTIVITIES`	Subsistence Crop Farmer Growing Crops for Sale

Enumerators assign ISCO and ISIC codes based on the descriptions. The first digit refers to the most general category, while the fourth digit indicates the most granular. Note that in this case the ISCO third and fourth digits capture identical information, but this is not always the case. For ISIC, the first digit is N/A because the first two digits together describe the most general category.

2.2. Classification Challenges

Converting a description into ISCO and ISIC codes is complex as many codes are nuanced and enumerators face severe time constraints. ZamStats officials overseeing the LFS have raised these concerns during discussions with the International Growth Centre (IGC). To better understand these issues, we monitored the survey process for one week across two enumeration areas. During monitoring and subsequent discussions with the LFS team, three issues have emerged:

- **Complex Classifications:** There are 436 possible ISCO codes, with a codebook spanning 433 pages, and 419 possible ISIC codes, with a codebook spanning 306 pages. Enumerators are unlikely to recall each section of the codebook nor have time to parse the entire document.
- **Incomplete Descriptions:** Enumerators often record short written descriptions of responses, even in cases where the respondent goes into great detail. This makes it more difficult to assign accurate codes later. For example, the caption in Table 1 highlights that enumerators often report nearly identical details for the title, description, and main activities, although these sections are distinct.
- **Spelling Mistakes:** To retrieve the correct code, enumerators can search their electronic codebook using keywords from the description. In cases where they make severe spelling mistakes, enumerators will fail to retrieve the correct code, leading to mistaken assignment.

These challenges place considerable pressure on enumerators to complete ISCO and ISIC codes accurately and on time. As an alternative, we propose using LLMs to classify the responses.

3. Methods

In this section, we describe the task, dataset, LLM pipeline, scoring, and assumptions required for benchmark validity.

3.1. Task

The task is to classify descriptive responses into ISCO and ISIC codes. Each record contains three descriptive fields provided by respondents: Occupation Title (`D1_TITLE`), Main Tasks and Duties (`D1_DESC`), Main Activity, Goods, or Services (`D2_MAIN_ACTIVITIES`). The LLM's task is to interpret these free-text descriptions and return an ISCO or ISIC code, depending on the classification setting (for an example, see Table 1). The expected output is a JSON containing the ISCO or ISIC code.

3.2. Dataset

The dataset contains 1,000 observations sampled from the 2023 LFS. From 10,400 interviewed households, ZamStats shared a random sample of 1,059 observations containing the original descriptions and enumerator codes. Expert officials from ZamStats's LFS team, who oversee the survey process for the entire country, labelled these 1,059 observations. We deem these ground truth codes, which are independent of the enumerator codes. From this set, we drop 48 observations where the ISCO and/or ISIC codes are incomplete or contain invalid four-digit codes. After performing detailed error analysis on preliminary model evaluations, we exclude 11 observations to be used for few-shot prompting (see Appendix C.3). This leaves a dataset of 1,000 observations.

3.3. Classification Pipeline

Our classification pipeline uses RAG, few-shot prompting, and LLMs.

RAG We perform RAG using Google's 3072-dimensional Gemini embedding model (Google AI for Developers, 2025). All ISCO and ISIC codebook descriptions are encoded. At inference time, the respondent data is encoded, then the 20 most relevant documents are retrieved and supplied to the model as candidate options. In preliminary testing, we observe a number of cases where the true code is a 'not elsewhere classified' (n.e.c.) code (e.g., ISCO 1219: Business Services and Administration Managers Not Elsewhere Classified). These

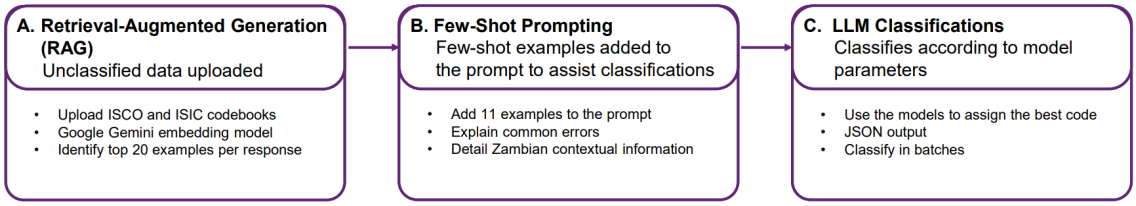


Figure 2: Methodology Overview. Once the raw survey data is inputted, we use **A.** RAG to identify the 20 most relevant codes, **B.** few-shot examples for maximum accuracy, and **C.** LLMs to identify the best code.

codebook descriptions typically lack the semantic similarity to reach the top 20 RAG examples. As such, the most similar n.e.c. code is always added to the 20 candidates if not already included. For additional methodological details and RAG performance results, please see Appendix C.4.

Few-Shot Prompting We also harness few-shot prompting. Each LLM is shown 11 hand-picked examples that illustrate common sources of error. These include vague job descriptions and jobs specific to the Zambian context, where LLMs might have less background knowledge. Each few-shot example is accompanied with an explanation of the correct code. The few-shot examples and explanations can be viewed in Appendix C.

Models We evaluate GPT-5 (OpenAI, 2025a), GPT-5 mini (OpenAI, 2025b), and GPT-5 Nano (OpenAI, 2025c) to test OpenAI's latest models, as well as legacy models GPT-4 Turbo (OpenAI, 2024) and GPT-4.1 (OpenAI, 2025d) to measure performance improvements. From Google, we evaluate flagship model Gemini 2.5 Pro (Google, 2025), and we include Meta's Llama 3.1 70B (Meta AI, 2024) to test an open source LLM.

3.4. Scoring

To score LLM-generated classifications against the ground truth codes, we employ two metrics: Exact Match and Partial Match. In both cases, we report LLM performance against enumerator codes as a human baseline.

Exact Match This evaluates whether the LLM-generated code is identical to the ground truth across all four digits. A response is scored as correct only if a perfect match occurs. This avoids introducing human judgment into the scoring process; classifications either match the ground truth or they do not. However, this approach depends on the availability of high-quality ground truth labels, as discussed in Appendix A.

Partial Match Because exact match fails to reward partially correct answers (Bean et al., 2024), we also allow for partial match on earlier digits. As higher digit accuracy depends on the accuracy in the preceding digits, the model must correctly predict digit n before it can receive credit for digit $n+1$. We present this as separate metrics for cumulative accuracy on digits one, two, and three.

3.5. Benchmark Validity

Rigorous LLM benchmarking requires considering the *validity* of the benchmark: whether it captures the real-world phenomenon it seeks to measure (Alaa et al., 2025). Our test assumes that the true jobs Zambians perform are faithfully represented by the ground truth ISCO and ISIC codes. This requires (i) that the ground truth ISCO and ISIC codes are correctly assigned; (ii) that the selected sample is representative of Zambia's labour force; and (iii) that ISCO and ISIC codes capture the full diversity of Zambia's occupations and industries.

Traditionally, this assumption has been decomposed into criterion validity (corresponding to (i) above), content validity (ii), and construct validity (iii) (Cronbach and Meehl, 1955), an approach that has been developed further by Messick (1998), Kane (2012), and Alaa et al. (2025).

Criterion validity This measures how well a benchmark task translates to effective performance on the real-world task. This is embodied by α, which is the conditional probability that accuracy on the benchmark task translates into accuracy on the real-world outcome (Alaa et al., 2025):

$$\alpha = P(\text{Correct on real-world outcome} \mid \text{Correct on benchmark task})$$

If $\alpha = 1$, the benchmark task perfectly predicts the real-world outcome. In our context, this requires the following assumption:

Assumption 1: Agreement between an LLM and the ground truth codes predicts correct LLM assignment of codes based on information presented in field interviews.

While the LLMs and enumerators have access to *written* descriptions, codes in the field are assigned based on *verbal* descriptions. This gap means that α is likely less than 1, as enumerators' written descriptions rarely capture the entire response. Compared against a hypothetically perfect interviewer and classifier with complete information from verbal responses, an LLM relying on written descriptions would likely fall short. To minimise this gap, we recommend that enumerators record responses in much greater detail (see Section 6).

Content Validity This evaluates the representativeness of a benchmark task. In our case, the ISCO and ISIC codes we classify must be representative of ISCO and ISIC codes that would be assigned if every Zambian answered the survey. Therefore, we assume:

Assumption 2: The classification sample is nationally representative of Zambia's labour force.

This entails two sub-assumptions: one for internal validity to the LFS, and one for external validity to other surveys.

Assumption 2.1: The classification sample is randomly drawn from the full set of LFS responses.

If the sample is not random, the model's performance on the benchmark task will not reflect performance on the wider LFS survey. From the 10,400 surveyed households in the 2023 LFS, ZamStats randomly selected 1,059 observations. We remove 48 examples of improper codes and 11 examples for few-shot prompting (see Section 3.2).

Assumption 2.2: The full set of LFS responses is nationally representative of Zambia's labour force.

ZamStats employs a two-stage stratified cluster sampling design, First, 520 enumeration areas (EAs) are selected with probability proportional to estimated size; and secondly, 20 households are selected within each sampled EA using systematic random sampling (Zambia Statistics Agency, 2024). EAs are allocated disproportionally across provinces, so sampling weights are required for national representativeness. Future updates to this benchmark will draw a weighted random sample.

Construct Validity This evaluates how well a test measures an underlying theoretical construct (Alaa et al., 2025). Our underlying constructs are occupations and industries, which are captured by ISCO and ISIC codes, respectively. We assume:

Assumption 3: ISCO codes capture the full range of occupations in Zambia and ISIC codes capture the full range of industries in Zambia.

As universal frameworks, ISCO and ISIC are not tailored to Zambia's labour market. This can limit their usefulness (International Labour Organization, 2008; United Nations Statistics Division, 2008), particularly in high-informality, low-specialisation contexts where workers may perform multiple job duties simultaneously. However, in the absence of country-specific codebooks, they are likely the best available frameworks in Zambia.

Illustrative Example Using the example of a subsistence crop farmer growing maize in Table 1, we note that:

- Criterion validity requires that the respondent in fact describes themselves as a subsistence crop farmer growing maize.

- Content validity implies that Zambia's economy contains subsistence crop farmers growing maize, and that this code and others in the sample are representative of the labour force.
- Construct validity means that ISCO code 6310 captures the real job of subsistence crop farmer, and ISIC code 0111 captures the growing of maize.

4. Results

Our results indicate that LLMs outperform the human baseline on exact match and partial match at low cost.

4.1. Exact Match

Figure 1 shows that LLMs outperform human enumerators on ISCO and ISIC classifications with statistical significance at the 99.9% confidence level. LLMs achieve up to 62.5% accuracy on ISCO (11.8 percentage points over enumerators) and 65.5% accuracy on ISIC (17.8 percentage points over enumerators).

4.2. Partial Match

Table 2 shows partial match results for ISCO. We find that LLMs outperform enumerators by up to 11.4 percentage points on the first digit, 10.0 percentage points on the first two digits, and 10.8 percentage points on the first three digits. All results are significant at the 99.9% confidence level.

Cumulative Digits	GPT–4.1	GPT–5	GPT–5 mini	GPT–5 Nano	Gemini 2.5 Pro
One	0.1140*** (0.0132)	0.1070*** (0.0138)	0.0940*** (0.0135)	0.0990*** (0.0131)	0.1010*** (0.0133)
Two	0.0970*** (0.0165)	0.1000*** (0.0163)	0.0850*** (0.0162)	0.0930*** (0.0154)	0.0890*** (0.0160)
Three	0.1080*** (0.0167)	0.1030*** (0.0170)	0.1000*** (0.0168)	0.0990*** (0.0163)	0.1060*** (0.0165)
Four	0.1180*** (0.0170)	0.1090*** (0.0172)	0.1020*** (0.0169)	0.1090*** (0.0165)	0.1120*** (0.0166)

Table 2: Accuracy Gain Over Enumerators (ISCO). Each cell shows the LLM's accuracy gain over enumerators for the ISCO classification task, with p-values computed using McNemar's test and Wald standard errors in parentheses. Performance results for Llama 3.1 70B and GPT-4 Turbo are included in Appendix B due to spacing constraints.
*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

Table 3 shows partial match results for ISIC. We find that LLMs outperform enumerators by up to 6.8 percentage points on the first two digits and 10.7 percentage points on the first three digits. All results are significant at the 95% confidence level, with a majority significant at the 99.9% confidence level.

Cumulative Digits	GPT–4.1	GPT–5	GPT–5 mini	GPT–5 Nano	Gemini 2.5 Pro
Two	0.0400** (0.0137)	0.0680*** (0.0130)	0.0660*** (0.0130)	0.0480*** (0.0129)	0.0590*** (0.0133)
Three	0.0510** (0.0166)	0.1040*** (0.0162)	0.1070*** (0.0159)	0.0810*** (0.0158)	0.0960*** (0.0163)
Four	0.1220*** (0.0182)	0.1580*** (0.0179)	0.1710*** (0.0176)	0.1380*** (0.0176)	0.1570*** (0.0178)

Table 3: Accuracy Gain Over Enumerators (ISIC). Each cell shows the LLM's accuracy gain over enumerators for the ISIC classification task, with p-values computed using McNemar's test and Wald standard errors in parentheses. Performance results for Llama 3.1 70B and GPT-4 Turbo, which were weaker on ISIC digits, are included in Appendix B due to spacing constraints.
*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

4.3. Cost

In low-resource settings such as Zambia, cost is a key consideration. Figure 3 shows exact match score plotted against the logged number of employed persons classified per US dollar. Please see Table 8 for exact details.

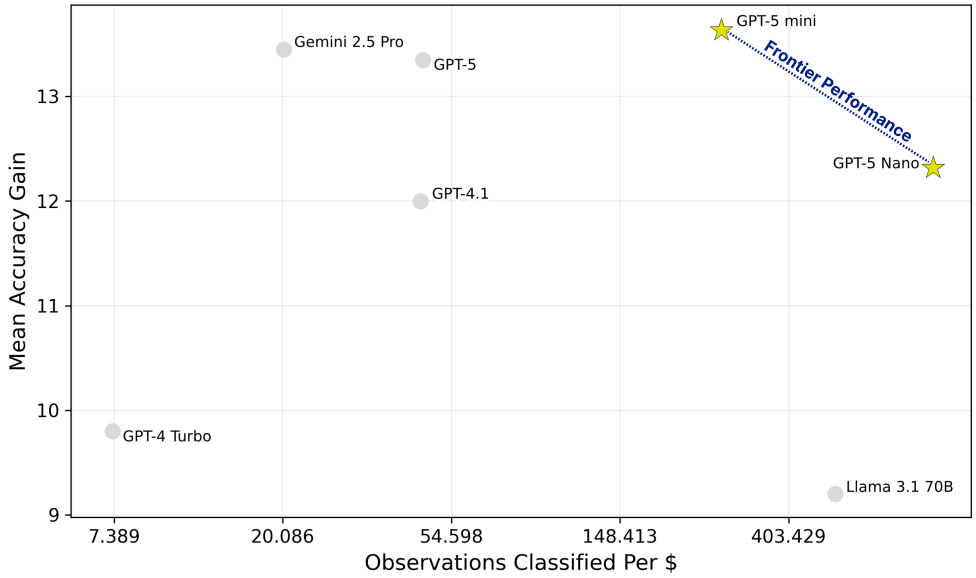


Figure 3: Cost Efficiency. This chart shows mean ISCO and ISIC fourth-digit classification accuracy gain against the enumerators plotted against observations classified per US dollar. The graph is plotted on a logarithmic scale. GPT-5 mini has the highest average accuracy, while GPT-5 Nano has the lowest cost per classification. These represent frontier performance.

5. Discussion

5.1. Model Recommendation

LLMs outperform enumerators on accuracy, with most doing so at relatively low cost. Figure 3 shows no discernible trend between increased costs and performance on this task. GPT-5 Nano obtains high performance at low cost, with exact match gains over enumerators surpassing 12 percentage points and a classification cost of $1.07 per 1,000 observations. Performance is maximised by GPT-5 mini, with exact match gains over enumerators exceeding 13 percentage points and a classification cost of $3.60 per 1,000 observations.

As shown in Appendix B, the differences between GPT-5 mini and GPT-5 Nano are insignificant on all ISCO digits but statistically significant ($p < 0.001$) on all ISIC digits.

Our recommendation depends on the preferences and resources of policymakers. In higher-resource contexts, or when sample sizes are small, we recommend GPT-5 mini. In lower-resource contexts, or where sample sizes are large, we recommend GPT-5 Nano. Given typical labour force participation rates in Zambia, policymakers could analyse an annual LFS sample of 10,400 households for under $10 using GPT-5 Nano or under $30 using GPT-5 mini. Performance plots for GPT-5 mini and GPT-5 Nano on all digits are available in Appendix B.

5.2. Error Analysis

After conducting extensive error analysis, we believe that our results represent the maximum possible performance on this task.

First, there are often incomplete descriptions that make identifying a single four-digit code impossible. For instance, ISIC code 4711 (Retail sale in non-specialized stores with food, beverages or tobacco predominating) and ISIC code 4721 (Retail sale of food in specialized stores) only differ in the degree of store specialisation.

With incomplete descriptions of a respondent's main activities (e.g., "Retail Sale of Meat Products"), the degree of store specialisation can only be inferred. In all sets of codes–enumerator, ground truth, and LLM we observe inconsistencies where these vague responses are arbitrarily assigned to different four-digit codes.

Second, we observe inaccuracies in ground truth data. During preliminary tests (performed before the release of GPT-5), we instruct GPT-4.1, o3 (OpenAI, 2025e), Gemini 2.5 Pro, Llama 3.1 70B, and GPT-4 Turbo to classify random sets of 100 observations without few-shot examples. After performing error analysis, we find instances where our knowledge of the Zambian context supports the LLM codes over the ground truth data.

5.3. Policy Implications

Speed of Production and Countercyclical Policy Officials from the ZamStats LFS team estimate that each classification takes around one minute for staff in Lusaka, and up to ten minutes for enumerators. With 10,400 responses annually, this can become a significant burden that trades off with other job responsibilities. Using conservative estimates of one minute per code for the LFS staff in Lusaka and three minutes for enumerators, we estimate over 43 working days saved for the team in Lusaka and 130 working days saved for enumerators, assuming a typical 8-hour workday.

This has the potential to increase the policy relevance of the official statistics by reducing the time to dissemination. While all LFS data collection for 2023 was finished by December 2023, the annual report for the survey was not produced until November 2024. By freeing labour power to be used on other outputs, our approach can speed up the statistical production process. Given that governments often conduct fiscal and monetary stabilization policies in response to labour market trends, increasing the speed at which reports are produced can help policymakers identify and rapidly respond to labour market shocks.

Sectoral Composition Improved estimates of ISCO and ISIC codes can reshape our understanding of occupation and industry distributions in Zambia. We classify the 7,701 employed persons from the 2022 LFS, the most recent from which we have a complete dataset, using GPT-5 mini as our top performing model. ISCO codes are mapped onto the 10 Major Groups (first digit categories), while ISIC codes are mapped onto the 21 Industry Sections (first two digit categories).

We use Bowker's test of symmetry (Bowker, 1948) to assess whether discrepancies between enumerator and GPT-5 mini classifications are systematically asymmetric across categories. Both occupation ($\chi^2(44) = 967.89$, $p < 0.001$) and industry ($\chi^2(131) = 528.08$, $p < 0.001$) classifications show statistically significant departures from symmetry. Then, we use McNemar's test (McNemar, 1947) with the Benjamini–Hochberg false discovery rate adjustment (Benjamini and Hochberg, 1995) to determine if category-level differences are statistically significant. For ISCO, we found systematic differences between GPT-5 mini and enumerators across every Major Group ($p < 0.05$), with most significant for $p < 0.001$. On ISIC, we found systematic differences between GPT-5 mini and enumerators on 11 of 21 Industry Sections ($p < 0.05$). A graph of ISCO results is available in Figure 4, while ISIC results are available in Figure 7 in Appendix B. Detailed results tables are available in Table 6 and Table 7.

The share of workers in services occupations is over 5 percentage points higher under GPT-5 mini's classifications (32.8% against 27.7%). Simultaneously, the industrial shares of manufacturing (7.7% against 9.4%) and agriculture, forestry, and fishing (23.7% against 25.4%) are smaller (see Table 7). This has major implications for our understanding of Zambia's growth path and structural transformation. As Rodrik (2016) has argued, many developing countries are turning into service economies without manufacturing-led growth. Our results suggest that Zambia may be experiencing this phenomenon more than previously imagined, a finding critical for industrial and other sector-specific policies.

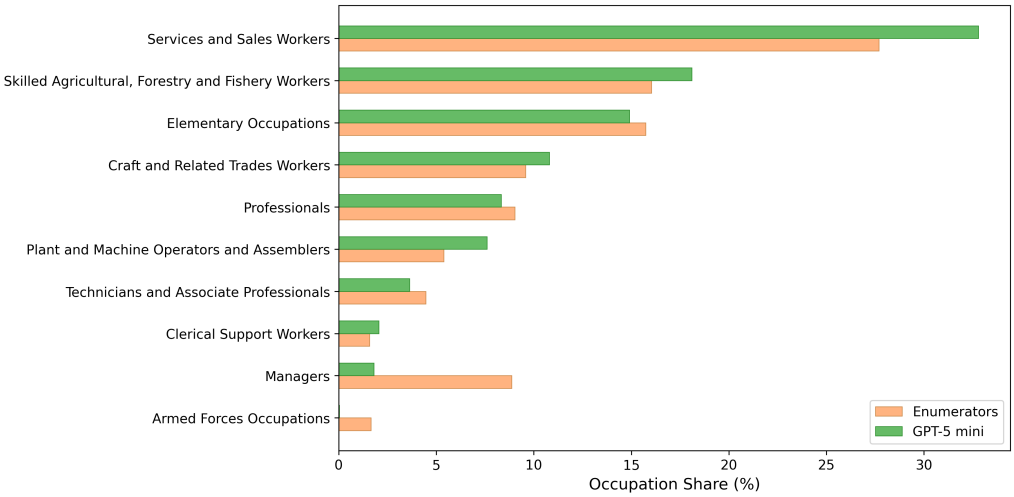


Figure 4: Enumerator vs LLM Classification Shares: Occupations. Shares across occupation Major Groups assigned by human enumerators and GPT-5 mini. All differences are statistically significant for $p < 0.05$. Confidence intervals are omitted as they are based on the pairwise difference between GPT-5 mini and enumerators rather than the individual proportions.

6. Policy Recommendations

We offer three brief policy recommendations for national statistics agencies interested in pursuing our approach.

6.1. Implementation

First, our LLM pipeline can be implemented via a Graphical User Interface (GUI), which we plan to open source. After inputting an Excel spreadsheet with the necessary input columns, the user may select their desired LLM to run the classifications. No coding is required. We presented this approach to ZamStats in July 2025 and are targeting full implementation by the end of 2025. For more information on the GUI, please see Appendix F.

6.2. Enumerator Training

To improve classification accuracy further, detailed enumerator descriptions are of paramount importance (see Section 5.1). Given that our approach is likely to reduce or eliminate the time enumerators spend on ISCO and ISIC classification, national statistics agencies might encourage enumerators to instead record information in as much detail as possible.

In a Zambian context, we recommend that enumerators recording the occupation title and main tasks and duties for ISCO classification note the respondent's role (independent operator, manager, etcetera). For farmers, we recommend that enumerators record if farming is primarily subsistence or for sale, as well as the primary type of crop cultivated.

When recording the main, activity, goods, or services for ISIC classification, we recommend that enumerators note the degree of specialisation (e.g., a specialised or non-specialised store), as well as the location where work is performed (market, shop, grocery store, roadside, etcetera).

6.3. Census Classification

Our approach could be scaled to Zambia's 2022 Census (Zambia Statistics Agency, 2022), a national survey of Zambia's over four million households. With such a comprehensive sample, assigning ISCO and ISIC codes to Census job responses has been too resource-intensive for human enumerators. Using GPT-5 Nano, we estimate

that every employed Zambian in the Census could be assigned ISCO and ISIC codes for around $4,250.[1]

7. Related work

ISCO and ISIC Classification Previous studies have proposed natural language processing methods to automate ISCO and ISIC classification. However, these typically focus on subsets of the data, such as responses from parents (Duckworth and Fraillon, 2023), adolescents (Safikhani et al., 2023), and environmental activities (Li et al., 2024), or classify business entities rather than workers themselves (Béchara et al., 2022; Rizinski et al., 2023). Comprehensive analysis has previously been done in developed countries. Most notably, a RAG and LLM pipeline was used to classify labour market survey data in the UK (UK Government, 2025). This study is tailored to the British Standard Industrial Classification (SIC) system, and reports only "marginal" accuracy gains over existing approaches (UK Government, 2025). Similarly, Singapore's Department of Statistics uses LLMs to summarise job descriptions and an embedding-based classifier to assign codes (Lim et al., 2025). This method outperformed direct classification by GPT-4o-mini, but reliance on Singapore's unique classification system means that it is not applicable to other contexts. By contrast, our system is designed for broad applicability across developing countries and delivers substantial accuracy improvements over existing methods.

ISCO and ISIC in Africa To date, we believe that machine learning approaches to ISCO and/or ISIC classification have only been publicly documented in four African countries beyond Zambia. In Egypt, previous approaches classified online job postings, rather than nationally representative survey data (El-Lawah et al., 2025). In Tunisia, researchers trained machine learning models (the most successful based on logistic regression or a decision tree classifier) to assign ISIC codes based on enumerator data (Mansour and Al Taharwah, 2023). Beyond the limitations of traditional machine learning models, training on enumerator data risks reproducing the same enumerator biases that we document. With Ghanaian and South African data, researchers classified businesses into ISIC codes, but the use of legacy encoder-decoder models and focus on corporate entities (rather than workers) makes our project comparatively distinct (Béchara et al., 2022; Naveed et al., 2025).

Arguably, the most similar approach was the Ghana Statistics Service using a RoBERTa model (Smeets and Baako-Amponsah, 2023). However, our use of state-of-the-art LLMs is likely to yield improved performance on national language understanding and generation tasks (Naveed et al., 2025). For these reasons, we contend that our approach is the most advanced to date in an African context and set to become the first sustainably implemented across an African statistics agency.

LLM Benchmarking Our work also contributes to the wider field of LLM benchmarking and evaluation. A benchmark is a structured way of evaluating performance in a specific domain (Ruder, 2021). Benchmarks are used to track progress in high-level domains such as natural language understanding (Wang et al., 2018; Rajpurkar et al., 2016), maths (Cobbe et al., 2021; Hendrycks et al., 2021; Glazer et al., 2024), and reasoning (Rein et al., 2024; Hendrycks et al., 2021; Bean et al., 2024; Khouja et al., 2025). Recently, more applied benchmarks have been created for specific economic fields, such as coding (Chen et al., 2021; Jimenez et al., 2024), healthcare (Jin et al., 2021; Arora et al., 2025), and autonomously conducting academic research (Starace et al., 2025; Xiang et al., 2025). Our work contributes to this literature by evaluating performance on a specific task: labour force survey classification. We also rely heavily on papers outlining best practices in benchmarking (Biderman et al., 2024).

LLMs in Low-Resource Settings Our pipeline uses LLMs trained on corpora consisting of mostly English text and reflecting norms prevalent in developed countries. Furthermore, models are post-trained with reinforcement learning with human feedback to default to the norms of developed countries (Ouyang et al., 2022). A significant literature studies how LLMs generalise to low-resource settings such as non-English speaking countries (Adelani et al., 2024) and those with different cultures (AlKhamissi et al., 2024; Singh et al., 2025). In our study, we identify notable cases where the LLMs provide different classifications to the Zambian enumerators because they assume a developed country viewpoint and lack the Zambian context. We report examples of these in Appendix C.2.

[1] In the 2023 LFS, the total employed population is 3,980,733 (Zambia Statistics Agency, 2024). At a classification cost of $0.00107 per observation using GPT-5 Nano (Table 8), total classification costs equal $4,259.

8. Conclusion

We introduce a low-cost, high-accuracy solution to ISCO and ISIC coding. Our LLM-based method significantly outperforms human enumerators with exact match accuracy gains of up to 17.1 percentage points and costs as low as $1.07 per 1,000 records. Cost-effective models such as GPT-5 Nano perform excellently on performance metrics, demonstrating the potential of LLMs to aid resource-constrained national statistics agencies.

Our approach enables faster countercyclical economic measures and better targeting of sectoral policies in Zambia. If adopted widely, similar benefits could be available to other African countries and in separate survey contexts.

Author Contributions

Tyler Rossow wrote the paper, led the policy and economics applications, provided development economics expertise throughout the project, and assisted with coding.
Tobias Edison built the GUI, configured the API setup, assisted with coding, and contributed to paper writing.
Rory Hardie developed the original project proposal, supported the policy and economics applications, monitored the survey process in the field, and contributed to paper writing.
Harry Mayne supervised the methodology design, wrote the RAG pipeline and main LLM inference code, provided LLM expertise throughout the project, and contributed to paper writing.

References

D. I. Adelani, H. Liu, X. Shen, N. Vassilyev, J. O. Alabi, Y. Mao, H. Gao, and E.-S. A. Lee. SIB-200: A Simple, Inclusive, and Big Evaluation Dataset for Topic Classification in 200+ Languages and Dialects. In Y. Graham and M. Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta, Mar. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.14. URL https://aclanthology.org/2024.eacl-long.14/.

A. Alaa, T. Hartvigsen, N. Golchini, S. Dutta, F. Dean, I. D. Raji, and T. Zack. Medical large language model benchmarks should prioritize construct validity. *arXiv preprint arXiv:2503.10694*, 2025.

B. AlKhamissi, M. ElNokrashy, M. Alkhamissi, and M. Diab. Investigating Cultural Alignment of Large Language Models. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12404–12422, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.671. URL https://aclanthology.org/2024.acl-long.671/.

R. K. Arora, J. Wei, R. S. Hicks, P. Bowman, J. Quiñonero-Candela, F. Tsimpourlas, M. Sharman, M. Shah, A. Vallone, A. Beutel, J. Heidecke, K. Singhal, et al. HealthBench: Evaluating Large Language Models Towards Improved Human Health. *arXiv preprint arXiv:2505.08775*, 2025.

A. M. Bean, S. Hellsten, H. Mayne, J. Magomere, E. Chi, R. Chi, S. Hale, and H. R. Kirk. LINGOLY: A benchmark of olympiad-level linguistic reasoning puzzles in low resource and extinct languages. *Advances in Neural Information Processing Systems*, 37:26224–26237, 2024.

H. Béchara, R. Zhang, S. Yuan, and S. Jankin. Applying NLP Techniques to Classify Businesses by their International Standard Industrial Classification (ISIC) Code. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 3472–3477. IEEE, 2022.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

S. Biderman, H. Schoelkopf, L. Sutawika, L. Gao, J. Tow, B. Abbasi, A. F. Aji, P. S. Ammanamanchi, S. Black, J. Clive, A. DiPofi, J. Etxaniz, B. Fattori, J. Z. Forde, C. Foster, J. Hsu, M. Jaiswal, W. Y. Lee, H. Li, C. Lovering, N. Muennigho, E. Pavlick, J. Phang, A. Skowron, S. Tan, X. Tang, K. A. Wang, G. I. Winata, F. Yvon, and A. Zou. Lessons from the Trenches on Reproducible Evaluation of Language Models. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2405.14782. URL https://doi.org/10.48550/arXiv.2405.14782.

A. H. Bowker. A test for symmetry in contingency tables. *Journal of the American Statistical Association*, 43 (244):572–574, 1948.

M. Chen, J. Tworek, H. Jun, Q. Yuan, , et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021.

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training Verifiers to Solve Math Word Problems, 2021. URL https://arxiv.org/abs/2110.14168.

L. J. Cronbach and P. E. Meehl. Construct validity in psychological tests. *Psychological bulletin*, 52(4):281, 1955.

D. Duckworth and J. Fraillon. Improving Parental Occupation Coding Procedures with AI. Technical report, International Association for the Evaluation of Educational Achievement (IEA), 2023.

A. El-Lawah, A. Habashy, Y. Nasr, A. Dawoud, S. Samir, and A. Saleh. An AI-Driven Lens on the Demand Side of the Egyptian Labor Market (2021–to Date) | Part II: An Agentic-AI System for ISCO-08 Occupational Classification. Working Paper ECES-WP242-E, Egyptian Center for Economic Studies (ECES), Cairo, Egypt, July 2025. URL https://eces.org.eg/en/an-ai-driven-lens-on-the-demand-side-of-the-egyptian-labor-market-2021-to-datepart-ii-an-a

E. Glazer, E. Erdil, T. Besiroglu, D. Chicharro, E. Chen, A. Gunning, C. F. Olsson, J.-S. Denain, A. Ho, E. de Oliveira Santos, O. Järviniemi, M. Barnett, R. Sandler, M. Vrzala, J. Sevilla, Q. Ren, E. Pratt, L. Levine, G. Barkley, N. Stewart, B. Grechuk, T. Grechuk, S. V. Enugandla, and M. Wildon. FrontierMath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI, 2024. URL https://arxiv.org/abs/2411.04872.

Google. Gemini models. https://ai.google.dev/gemini-api/docs/models, 2025. Accessed: 2025-08-20.

Google AI for Developers. Embeddings — Gemini API: Model versions. https://ai.google.dev/gemini-api/docs/embeddings#model-versions, 2025. Google. Last updated 2025-08-21. Accessed 2025-08-29.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

International Labour Organization. International Standard Classification of Occupations (ISCO). https://ilostat.ilo.org/methods/concepts-and-definitions/classification-occupation/, 2008. Accessed: 2025-08-26.

C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. R. Narasimhan. SWE-bench: Can Language Models Resolve Real-world Github Issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits. What Disease Does This Patient Have? A Large-Scale Open Domain Question Answering Dataset from Medical Exams. *Applied Sciences*, 11(14), 2021. ISSN 2076-3417. doi: 10.3390/app11146421. URL https://www.mdpi.com/2076-3417/11/14/6421.

M. Kane. All validity is construct validity. Or is it? *Measurement: Interdisciplinary Research & Perspective*, 10 (1-2):66–70, 2012.

J. Khouja, K. Korgul, S. Hellsten, L. Yang, V. Neacsu, H. Mayne, R. Kearns, A. Bean, and A. Mahdi. LINGOLY-TOO: Disentangling Reasoning from Knowledge with Templatised Orthographic Obfuscation. *arXiv preprint arXiv:2503.02972*, 2025.

X. Li, L. Zhao, J. Ren, Y. Sun, C. F. Tan, Z. Yeo, and G. Xiao. A Unified Framework to Classify Business Activities into International Standard Industrial Classification through Large Language Models for Circular Economy. In *2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1422–1426. IEEE, 2024.

J. Lim, L. Ng, and R. Erh. Automating Classification with DOS Intelligent Classification Engine (DICE). In *Generative AI and Official Statistics Workshop 2025, Conference of European Statisticians*, Geneva, Switzerland, May 2025.

J. Mansour and F. Al Taharwah. Using Machine Learning Approaches for Economic Classification Based on Arabic Textual Descriptions, 2023.

Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

S. Messick. Test validity: A matter of consequence. *Social indicators research*, 45(1):35–44, 1998.

Meta AI. Introducing Llama 3.1: Our most capable models to date. https://ai.meta.com/blog/meta-llama-3-1/, 2024. Accessed: 2025-08-20.

E. Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

J. T. Newsom. Matched Pairs Analysis. https://web.pdx.edu/~newsomj/cdaclass/ho_matched.pdf, 2021. Accessed 29 August 2025.

OpenAI. GPT-4 Turbo. https://platform.openai.com/docs/models/gpt-4-turbo, 2024. Accessed: 2025-08-20.

OpenAI. GPT-5 is here. https://openai.com/gpt-5/, 2025a. Accessed: 2025-08-20.

OpenAI. GPT-5 Mini. https://platform.openai.com/docs/models/gpt-5-mini, 2025b. Accessed: 2025-08-20.

OpenAI. GPT-5 nano. https://platform.openai.com/docs/models/gpt-5-nano, 2025c. Accessed: 2025-08-20.

OpenAI. GPT-4.1. https://platform.openai.com/docs/models/gpt-4.1, 2025d. Accessed: 2025-08-20.

OpenAI. Introducing OpenAI o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/, 2025e. Accessed: 2025-08-21.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *EMNLP*, 2016.

D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

M. Rizinski, A. Jankov, V. Sankaradas, E. Pinsky, I. Miskovski, and D. Trajanov. Company classification using zero-shot learning. *arXiv preprint arXiv:2305.01028*, 2023.

D. Rodrik. Premature deindustrialization. *Journal of economic growth*, 21(1):1–33, 2016.

S. Ruder. Challenges and Opportunities in NLP Benchmarking. https://www.ruder.io/nlp-benchmarking/, Aug. 2021. Blog post.

P. Safikhani, H. Avetisyan, D. Föste-Eggers, and D. Broneske. Automated occupation coding with hierarchical features: A data-centric approach to classification with pre-trained language models. *Discover Artificial Intelligence*, 3(1):6, 2023.

S. Singh, A. Romanou, C. Fourrier, D. I. Adelani, J. G. Ngui, D. Vila-Suero, P. Limkonchotiwat, K. Marchisio, W. Q. Leong, Y. Susanto, R. Ng, S. Longpre, S. Ruder, W.-Y. Ko, A. Bosselut, A. Oh, A. Martins, L. Choshen, D. Ippolito, E. Ferrante, M. Fadaee, B. Ermis, and S. Hooker. Global MMLU: Understanding and Addressing Cultural and Linguistic Biases in Multilingual Evaluation. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.919. URL https://aclanthology.org/2025.acl-long.919/.

L. Smeets and J. Baako-Amponsah. Enhancing Text Classification for Accurate Statistics: Leveraging LLM at the Ghana Statistics Service. Presentation at UNECA African Centre for Statistics (ACS) "Stats Talk" webinar, Sept. 2023. URL https://www.uneca.org/eca-events/sites/default/files/resources/documents/acs/stats-talk/2023-09-29/llms_laurent_josephine.pdf.

G. Starace, O. Jaffe, D. Sherburn, J. Aung, J. S. Chan, L. Maksin, R. Dias, E. Mays, B. Kinsella, W. Thompson, J. Heidecke, A. Glaese, and T. Patwardhan. PaperBench: Evaluating AI's Ability to Replicate AI Research. *arXiv preprint arXiv:2504.01848*, 2025.

UK Government. Artificial Intelligence Playbook for the UK Government, February 2025. URL https://www.gov.uk/government/publications/ai-playbook-for-the-uk-government/artificial-intelligence-playbook-for-the-uk-government-html. Accessed: 20/08/2025.

United Nations Statistics Division. *International Standard Industrial Classification of All Economic Activities (ISIC), Revision 4*. Statistical Papers, Series M, No. 4, Rev. 4. United Nations, New York, 2008. URL https://unstats.un.org/unsd/publication/seriesm/seriesm_4rev4e.pdf.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In T. Linzen, G. Chrupała, and A. Alishahi, editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Y. Xiang, H. Yan, S. Ouyang, L. Gui, and Y. He. SciReplicate-Bench: Benchmarking LLMs in Agent-driven Algorithmic Reproduction from Research Papers. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=8LoPjpvWde.

Zambia Statistics Agency. 2022 Census of Population and Housing: Preliminary Report. Preliminary report, Zambia Statistics Agency, Lusaka, Zambia, Dec. 2022. URL https://www.zamstats.gov.zm/wp-content/uploads/2023/12/2022-Census-of-Population-and-Housing-Preliminary.pdf.

Zambia Statistics Agency. 2023 Labour Force Survey (LFS) Report. Technical report, Zambia Statistics Agency, Lusaka, Zambia, Nov. 2024. URL https://www.mlss.gov.zm/wp-content/uploads/2024/11/2023-Labour-Force-Survey-Report-03112024-1.pdf.

A. Limitations

Limited Geographic Scope Our analysis focuses exclusively on Zambian LFS data. While we expect our results to generalise to developing countries with a comparable economic structure, further research is required.

Prompt Engineering Our study employs a standard prompting approach without exploring alternative prompt designs. It is possible that better prompt wording or more comprehensive few-shot learning strategies might lead to higher final performance. We note that the best performing models, e.g. the GPT-5 family, use inference-time compute to generate long chain-of-thoughts before returning an answer. As a result, it was not necessary to test chain-of-thought prompting.

Limited Baseline Comparisons While we establish a human baseline using enumerator performance, we do not test other automated classification methods, such as traditional machine learning approaches. While we expect our methods to significantly outperform these methods, future work could conduct comprehensive analysis to contextualise our findings.

Replication Considerations We primarily consider proprietary LLMs since they outperform open-source models in other benchmark tasks. This limits the reproducibility of our results because the LLM systems offered by the APIs may change. Model weights, training procedures, and updates are not publicly accessible for independent verification. To limit this, we provide the API endpoints used for all experiments in Section E.

Sample Size Our sample of 1,000 is sufficient to achieve statistically significant results comparing LLMs and enumerators, but the statistical difference between LLMs is limited. With a larger dataset, systematic differences between models are likelier to become evident. Similarly, the dataset size limits our ability to characterise model accuracy across rare occupation and industry categories.

Additionally, a smaller dataset subtly increases the risk of overfitting. While our pipeline is never trained on the data, we run preliminary tests before the final scoring round. Model performance may appear stronger on a small, more homogeneous sample than on the full dataset or in other survey contexts. Future work should therefore validate the approach on the complete dataset and on larger, more diverse LFS data to ensure robustness and generalisability.

Ground Truth Data Our results suggest that frontier LLMs achieve up to 66% exact match accuracy on these classification tasks. Extensive error analysis suggests that the remaining gap is largely caused by ambiguous respondent data (see Appendix C.2). We suspect that further increases in LLM intelligence would not correspond to higher performance on the benchmark. In the future, we hope to develop a gold standard ground truth dataset with detailed responses and ground truth codes that have undergone multiple annotation rounds.

B. Results

B.1. Llama 3.1 70 B and GPT-4 Turbo

Digits	Llama 3.1 (ISCO)	GPT-4 Turbo (ISCO)	Llama 3.1 (ISIC)	GPT-4 Turbo (ISIC)
One	0.1020*** (0.0133)	0.0980*** (0.0138)	–	–
Two	0.1030*** (0.0159)	0.0930*** (0.0159)	0.0050 (0.0142)	0.0330* (0.0138)
Three	0.0950*** (0.0164)	0.0990*** (0.0165)	0.0230 (0.0169)	0.0390* (0.0168)
Four	0.0940*** (0.0165)	0.1050*** (0.0169)	0.0900*** (0.0187)	0.0910*** (0.0183)

Table 4: Accuracy Gain Over Enumerators. Each cell shows the LLM's accuracy gain over enumerators for the LFS classification task, with p-values computed using McNemar's test and Wald standard errors in parentheses. While column headers are abridged for brevity, digit accuracy is reported cumulatively and Llama 3.1 70B is used, as throughout the paper.
*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

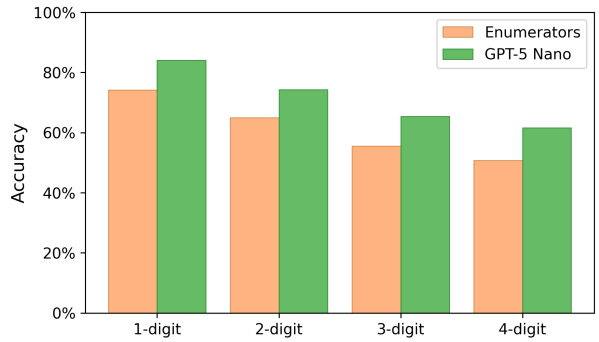
B.2. GPT-5 mini and GPT-5 Nano

Cumulative Digits	ISCO	ISIC
One	-0.005 (0.0090)	–
Two	-0.008 (0.0101)	0.018* (0.0075)
Three	0.001 (0.0102)	0.026** (0.0084)
Four	-0.007 (0.0102)	0.033*** (0.0093)

Table 5: Accuracy Difference (Mini Less Nano). Each cell shows the paired accuracy difference between GPT–5 mini and GPT–5 Nano, with p-values computed using McNemar's test and Wald standard errors in parentheses. ISIC results begin on the second digit as the first two digits combined are the most granular level.
*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

B.3. GPT-5 Nano

A. ISCO



B. ISIC

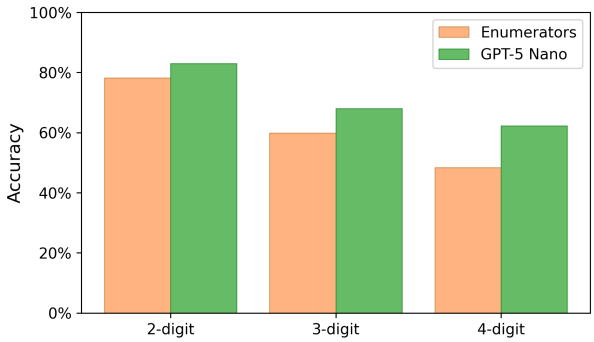
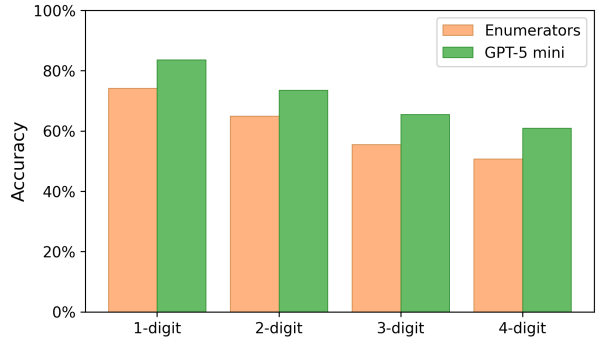


Figure 5: GPT-5 Nano ISCO and ISIC Evaluation. This figure shows the performance of GPT-5 Nano, our most cost-effective model, against human enumerators on the **A.** International Standard Classification of Occupations (ISCO) and **B.** International Standard Industrial Classification (ISIC) classification task. All results are significant at the 99.9% confidence level. Error bars are omitted as confidence intervals are only informative on the pairwise difference between the LLM and enumerators, rather than on the individual proportions.

B.4. GPT-5 mini

A. ISCO



B. ISIC

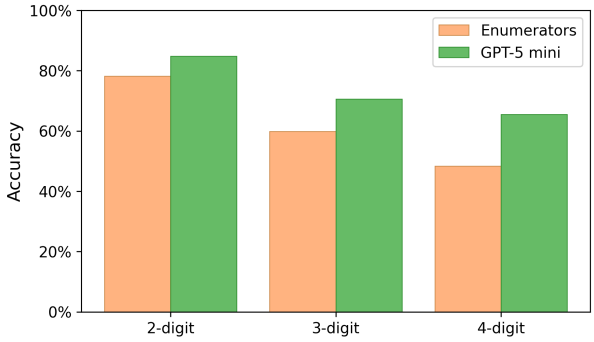


Figure 6: GPT-5 mini ISCO and ISIC Evaluation. This figure shows the performance of GPT-5 mini, our best performing model, against human enumerators on the **A.** International Standard Classification of Occupations (ISCO) and **B.** International Standard Industrial Classification (ISIC) classification task. All results are significant at the 99.9% confidence level. Error bars are omitted as confidence intervals are only informative on the pairwise difference between the LLM and enumerators, rather than on the individual proportions.

B.5. Industry Distribution

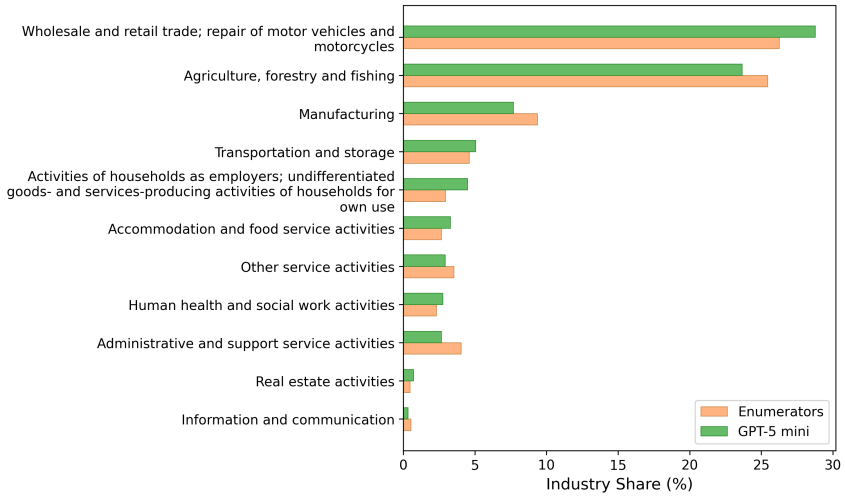


Figure 7: Enumerator vs LLM Classification Shares: Industries. Shares across Industry Sections assigned by human enumerators and GPT-5 mini. All differences depicted are statistically significant for $p < 0.05$. Confidence intervals are omitted as they are based on the pairwise difference between GPT-5 mini and enumerators rather than the individual proportions. Please note that there are ten Industry Sections omitted from this chart for which differences are not statistically significant.

B.6. Economic Composition

Occupation Major Group	b	c	Discordant	LLM Share	Enumerator Share	LLM Less Enumerator
Services and Sales Workers	395	788	1183	0.328	0.277	0.051*** (0.0044)
Plant and Machine Operators and Assemblers	47	217	264	0.076	0.054	0.022*** (0.0021)
Skilled Agricultural, Forestry and Fishery Workers	122	282	404	0.181	0.160	0.021*** (0.0026)
Craft and Related Trades Workers	168	261	429	0.108	0.096	0.012*** (0.0027)
Clerical Support Workers	51	88	139	0.021	0.016	0.005** (0.0015)
Professionals	159	105	264	0.083	0.090	-0.007** (0.0021)
Technicians and Associate Professionals	210	145	355	0.036	0.045	-0.008*** (0.0024)
Elementary Occupations	501	436	937	0.149	0.157	-0.008* (0.0040)
Armed Forces Occupations	126	1	127	0.0003	0.016	-0.016*** (0.0015)
Managers	595	51	646	0.018	0.089	-0.071*** (0.0032)

Table 6: Occupation Major Group Analysis. b = enumerator assigns group, GPT-5 mini does not; c = GPT-5 mini assigns group, enumerator does not. Wald standard errors are shown in parentheses.
Adjusted p-values using the Benjamini–Hochberg False Discovery Rate (FDR) procedure: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

Industry Section	b	c	Discordant	LLM Share	Enumerator Share	LLM Less Enumerator
Wholesale and retail trade; repair of motor vehicles	240	433	673	0.288	0.263	0.025*** (0.0034)
Activities of households as employers	33	152	185	0.045	0.029	0.015*** (0.0018)
Accommodation and food service activities	80	127	207	0.033	0.027	0.006** (0.0019)
Human health and social work activities	35	70	105	0.028	0.023	0.005** (0.0013)
Transportation and storage	29	63	92	0.050	0.046	0.004** (0.0012)
Construction	44	66	110	0.049	0.046	0.003 (0.0014)
Real estate activities	1	20	21	0.007	0.005	0.002*** (0.0006)
Education	15	26	41	0.061	0.059	0.001 (0.0008)
Electricity, gas, steam, air conditioning supply	4	9	13	0.002	0.002	0.001 (0.0005)
Water supply; sewerage, waste management	8	5	13	0.002	0.003	-0.0004 (0.0005)
Activities of extraterritorial orgs and bodies	5	2	7	0.0003	0.001	-0.0004 (0.0003)
Financial and insurance activities	21	17	38	0.011	0.012	-0.0005 (0.0008)
Professional, scientific, technical activities	44	38	82	0.008	0.008	-0.0008 (0.0012)
Arts, entertainment, recreation	27	20	47	0.004	0.005	-0.0009 (0.0009)
Mining and quarrying	28	15	43	0.019	0.021	-0.0017 (0.0009)
Public admin and defence	92	79	171	0.020	0.022	-0.0017 (0.0017)
Information and communication	26	10	36	0.003	0.005	-0.0021* (0.0008)
Other service activities	90	43	133	0.029	0.035	-0.0061*** (0.0015)
Administrative and support services	191	85	276	0.027	0.040	-0.0138*** (0.0022)
Manufacturing	310	180	490	0.077	0.094	-0.0169*** (0.0029)
Agriculture, forestry and fishing	259	122	381	0.237	0.254	-0.0178*** (0.0025)

Table 7: Industry Section Analysis. b = enumerator assigns section, GPT-5 mini does not; c = GPT-5 mini assigns section, enumerator does not. Wald standard errors are shown in parentheses. Some Industry Section names are abbreviated due to spacing constraints.
Adjusted p-values using the Benjamini–Hochberg False Discovery Rate (FDR) procedure: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

C. Methods

C.1. Significance Tests

All evaluation results are tested for statistical significance in accordance with the recommendation of Miller (2024). For the partial match figures in Tables 2, 3, 4, and 5, we use McNemar's test (McNemar, 1947) to compute p-values. This test is appropriate for paired binary outcomes, such as whether the LLM or the enumerator match the ground truth on the same survey response. Let b denote the number of cases where the LLM matches the ground truth and the enumerator does not, and c the number of cases where the enumerator matches the ground truth and the LLM does not. McNemar's chi-squared statistic is

$$\chi^2 = \frac{(|b-c|-1)^2}{b+c},$$

which under the null hypothesis $H_0 : P(\text{LLM matches ground truth}) = P(\text{Enumerator matches ground truth})$ is asymptotically distributed as χ^2 with one degree of freedom.[2]

In addition to p-values, we report the paired difference in proportions, $\Delta = (b-c)/n$, where $n = 1000$ is the total number of paired observations, together with Wald standard errors based on the large-sample multinomial variance:

$$SE(\Delta) \;=\; \sqrt{\frac{(p_b + p_c) - (p_b - p_c)^2}{n}}, \quad \text{with } p_b = b/n,\ p_c = c/n.$$

Wald standard errors are used because they provide a convenient large-sample approximation to the sampling distribution of Δ. In practice, they are commonly used in tandem with McNemar's test (Newsom, 2021).

For the occupation and industry composition charts in Figures 4 and 7, we use Bowker's test of symmetry (Bowker, 1948) to assess if category-level discrepancies are systematically asymmetric. Bowker's test is a multi-category generalization of McNemar's test: for a $K \times K$ contingency table of paired classifications, the null hypothesis is symmetry, i.e. $n_{ij} = n_{ji}$ for all $i \neq j$. The test statistic is

$$\chi^2 = \sum_{i<j} \frac{(n_{ij} - n_{ji})^2}{n_{ij} + n_{ji}},$$

which under H_0 is approximately χ^2-distributed with degrees of freedom equal to the number of non-redundant off-diagonal pairs.

On the category level for the occupation and industry composition charts, we use McNemar's test to evaluate whether differences are significant in individual categories. For each category k, we define $b =$ (LLM assigns k, enumerator does not assign k) and $c =$ (enumerator assigns k, LLM does not assign k). As above, we use McNemar's statistic with continuity correction and compute Wald standard errors.

Because we conduct multiple hypothesis tests (e.g., one McNemar test per category), naïve $p < 0.05$ thresholds would inflate the number of false positives. To address this, we apply the Benjamini–Hochberg (BH) procedure (Benjamini and Hochberg, 1995), which controls the false discovery rate, or the expected proportion of false rejections among all rejected hypotheses. This method inflates all p-values depending on the number of tests run and where the p-value for each test ranks, while preserving the overall ranked order of p-values.

Let $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(n)}$ denote the ordered set of n p-values. The BH adjustment computes adjusted values

$$p_{(i)}^{\text{adj}} \;=\; \min_{j \geq i} \left\{ \frac{n}{j}\, p_{(j)} \right\},$$

with monotonicity enforced so that $p_{(1)}^{\text{adj}} \leq p_{(2)}^{\text{adj}} \leq \cdots \leq p_{(n)}^{\text{adj}}$. Hypotheses are rejected if $p_{(i)}^{\text{adj}} \leq \alpha$, ensuring that the proportion of false discoveries is controlled at level α (e.g., 5%).

[2] For the comparison of GPT-5 mini and GPT-5 Nano (see Table 5), the null hypothesis is $H_0 : P(\text{GPT-5 mini matches ground truth}) = P(\text{GPT-5 Nano matches ground truth})$.

C.2. Error Analysis

During preliminary testing, we discover systematic errors made by LLMs. Generally, these are a consequence of models being trained on North American and European data, then applied to a developing country setting. Here, we highlight three representative examples. All examples were included in the few-shot prompt (see Appendix C.3) to encourage the model to take a Zambian view.

CASE C.2.1. Assumed economic sector

`D1_TITLE:`	Selling chicken for sale
`D1_DESC:`	Selling chicken
`D2_MAIN_ACTIVITIES:`	Selling chicken

In testing, models often assume this person is a fast-food worker at a chain restaurant. In Zambia, where there is a large informal sector, food sellers are likelier to sell on the street instead of in shops. This is an example of the developed country starting point causing misclassification.

CASE C.2.2. Developing country knowledge required

`D1_TITLE:`	Business lady (retailer)
`D1_DESC:`	Buying and selling of fried cassava
`D2_MAIN_ACTIVITIES:`	Buying and selling of fried cassava

Here, the model seems unaware of cassava, a staple root-vegetable in Zambia. This word is unlikely to appear frequently in developed country training text, so it is unsurprising that LLMs have limited knowledge.

CASE C.2.3. Zambia knowledge required

`D1_TITLE:`	Kachasu brewer
`D1_DESC:`	Selling kachasu at whole sale price
`D2_MAIN_ACTIVITIES:`	Selling kachasu beer at whole price

Classifying this example correctly requires significant knowledge of Zambia. The model needs to know (i) what kachasu is (an alcoholic beverage common in Zambia) (ii) that it is distilled (a different code) and (iii) that it is illegal in Zambia and therefore will not be sold in markets or stores. Most models fail at this task.

C.3. Prompts

In this section, we detail the prompts that we send to the LLM.

Listing 1: Main Prompt

```
"""
### TASK DESCRIPTION:

You need to classify respondent data using a list of codes to choose from. You will
    receive three key pieces of information:
- Occupational Title (D1_TITLE): The official or common name of the job.
  This information is noted in response to the question:
  "In his/her main job/business, what kind of work does (NAME) usually do?"
- Main Tasks and Duties (D1_DESC): A detailed description of what the job involves.
  This information is noted in response to the question:
  "In his/her main job/business, what kind of work does (NAME) usually do?"
- Main Activity, Goods, or Services (D2_MAIN_ACTIVITIES): The primary tasks and
    responsibilities. This information is noted in response to the question:
  "In (NAME) workplace, what kind of business activity is mainly carried out?"

### HOW YOU SHOULD SOLVE THIS TASK:

1. Read the respondent data carefully. This is provided at the end of the prompt.
    Pay attention to all information fields.
{priority_lines_string}
2. Look at the list of codes provided. Go through each code one-by-one. Consider
    how well the respondent data fits the code. You should consider the inclusion
    and exclusion criteria. Note that this is often difficult and subtle. You must
    reason carefully and with nuance to establish how well the respondent data fits
    each code.
3. Once you assess how suitable each code is, you should identify the most suitable
    code. This is often subtle. To help you do this, you should try to understand
    what the difference between the best codes is and how this affects the
    classification of the respondent data. Carefully read the definitions and the
    inclusion criteria.
4. Choose the optimal code and justify the choice of this code internally. Think:
    is this best?
5. Return the optimal code in the specified JSON format. The optimal code should
    always be 4-digits. Note that you will be rewarded for returning the correct
    code so think carefully about this.

### OUTPUT FORMAT:

Each object must have exactly these keys: {', '.join(json_keys)}.
{chr(10).join(code_instructions)}

### SUCCESSFUL EXAMPLES:

Here are some example of successfully assigned respondent data. Think about what
    makes these correct.
{few_shot_prompt}

### LIST OF CODES THAT YOU MUST CHOOSE FROM:

Go through these codes one-by-one and assess their compatibility with the
    respondent data (which will be provided later).
{"".join(individual_guidance_sections)}

### RESPONDENT DATA TO CLASSIFY:

Think about which code in the list is the best for the respondent data. Getting the
    correct code often requires you to understand the differences between similar
    codes. This is not always obvious.
In some cases, you will also need a detailed knowledge of Zambia, where informal
    self-employment is very common, maize is a staple food, and many sellers
    operate in stalls, markets, or on the street.
You are welcome to use other information you know about Zambia as needed.
"""
]
```

Explanatory Notes: ISCO

The following explanations are provided to LLMs to guide reasoning on the ISCO classification task.

Listing 2: Few-Shot Prompt ISCO

```
"""
    Read the explanations first, then study the single JSON array.

    Why these ISCO labels are correct (human notes; not JSON):
    - ID 250 (Soldier): A soldier is an armed forces officer, which is distinct
        from police. In Zambia, we can assume all soldiers are commissioned unless
        explicitly stated otherwise, so we assign 0110.
    - ID 742 (Chicken seller): In Zambia, where there is a large informal sector,
        food sellers are likelier to sell on the street instead of in shops, so we
        assign 5212. Stall and market salespersons, 5211, could also be plausible,
        but you should use the context of the response, the provided codebooks, and
        your knowledge of Zambia to make these judgments.
    - ID 256 (Kachasu brewer): Kachasu is a traditional distilled liquor which is
        homebrewed in Zambia. While it is illegal, sale remains common. Because of
        its illegal nature, it is likely that sales will not be in areas such as
        shops, markets, and the street, so we use 5249.
    - ID 748 (Farm worker): Crop farm labourers (9211) perform simple and routine
        tasks, usually on a farm that they do not own. As this person describes
        themselves as a farm worker, we can assume they do not own the farm and
        assign 9211.
    - ID 846 (Farm labourer): Crop farm labourers (9211) perform simple and routine
        tasks, usually on a farm that they do not own. As this person describes
        themselves as a farm labourer, we can assume they do not own the farm and
        assign 9211.
    - ID 226 (Marketer of vegetables): Because this person describes themselves as
        a marketer, and selling foods such as vegetables is common in Zambia's
        markets, we can assume they are a market salesperson and assign 5211.
    - ID 0 (Shift controller): This person is a supervisor, and 3121 specifically
        refers to mining, a large industry in Zambia.
    - ID 31 (Charcoal burner): Because this person identifies themselves as a
        charcoal burner and a salesperson, we can assume that they manufacture
        charcoal and sell it on the side of the road. Because charcoal is made by
        burning wood, 6210 is more accurate than other codes which describe
        activites such as logging. If this person only sold charcoal rather than
        also being a charcoal burner, we would use a code that only describes sales.
    - ID 687 (Retailer of fried cassava): Cassava is a starchy root vegetable
        widely cultivated in Zambia. Because Zambia has a high informality rate, we
        can assign 5212 for street food salespersons. Stall and market
        salespersons, 5211, could also be plausible, but you should use the context
        of the response, the provided codebooks, and your knowledge of Zambia to
        make these judgments.
    - ID 475 (Subsistence farmer): Subsistence farmers, even if they sell, are
        still considered for subsistence (beginning with 63) if production is
        primarily for own consumption. Per the official guidelines,  workers
        should be classified in Sub-major Group 63: Subsistence Farmers, Fishers,
        Hunters and Gatherers, if the main aim of production is to provide food,
        shelter, and other goods for consumption by members of the workers own
        household...Jobs should only be classified as market-oriented agricultural
        forestry, fishery or hunting if the main aim of the activity is to produce
        goods for the market.  While this person sells, their description of
        themselves as a subsistence farmer indicates that they likely primarily
        produce for their own consumption.
    - ID 913 (Farmer soybeans for sale): This person's activity is fully
        market-oriented (produced for sale, not primarily for own consumption with
        no mention of subsistence), so ISCO is 6111.

    """
]
```

Immediately after the explanatory notes, the following JSON array provides concrete examples formatted as input-output pairs.

Listing 3: Few-Shot ISCO Examples

```
[
  {
    "D1_TITLE": "SOLDIER",
    "D1_DESC": "PROTECTING THE COUNTRY",
    "D2_MAIN_ACTIVITIES": "PROTECTING THE COUNTRY",
    "ISCO_CODE_AI": "0110"
  },
  {
    "D1_TITLE": "SELLING CHICKEN FOR SALE",
    "D1_DESC": "SELLING OF CHICKEN",
    "D2_MAIN_ACTIVITIES": "SELLING OF CHICKEN",
    "ISCO_CODE_AI": "5212"
  },
  {
    "D1_TITLE": "KACHASU BREWER",
    "D1_DESC": "SELLING KACHASU AT WHOLE SALE PRICE",
    "D2_MAIN_ACTIVITIES": "SELLING KACHASU BEER AT WHOLE PRICE",
    "ISCO_CODE_AI": "5249"
  },
  {
    "D1_TITLE": "FARM WORKER",
    "D1_DESC": "PLANT MAIZE AND SOYBEANS",
    "D2_MAIN_ACTIVITIES": "GROW AND SELL MAIZE AND SOYBEANS",
    "ISCO_CODE_AI": "9211"
  },
  {
    "D1_TITLE": "FARM LABOURER",
    "D1_DESC": "CULTIVATING LAND FOR PLANTING MAIZE",
    "D2_MAIN_ACTIVITIES": "GROWING OF MAIZE",
    "ISCO_CODE_AI": "9211"
  },
  {
    "D1_TITLE": "MARKETER",
    "D1_DESC": "BUYING AND SELLING OF VEGETABLES",
    "D2_MAIN_ACTIVITIES": "SELLING OF VEGETABLES",
    "ISCO_CODE_AI": "5211"
  },
  {
    "D1_TITLE": "SHIFT CONTROLLER",
    "D1_DESC": "SUPERVISING SMELTER WORKS",
    "D2_MAIN_ACTIVITIES": "MINING OF COPPER",
    "ISCO_CODE_AI": "3121"
  },
  {
    "D1_TITLE": "CHARCOAL BURNER/SALES MAN",
    "D1_DESC": "SELLING CHARCOAL",
    "D2_MAIN_ACTIVITIES": "SALE OF CHARCOAL",
    "ISCO_CODE_AI": "6210"
  },
  {
    "D1_TITLE": "BUSINESS LADY (RETAILER)",
    "D1_DESC": "BUYING AND SELLING OF FRIED CASSAVA",
    "D2_MAIN_ACTIVITIES": "BUYING AND SELLING OF FRIED CASSAVA",
    "ISCO_CODE_AI": "5212"
  },
  {
    "D1_TITLE": "SUBSISTENCE FARMER",
    "D1_DESC": "SELL MAIZE GROW AND SELL MAIZE SOYBEANS",
    "D2_MAIN_ACTIVITIES": "SELL MAIZE GROW AND SELL MAIZE SOYBEANS",
    "ISCO_CODE_AI": "6310"
  },
  {
    "D1_TITLE": "FARMER",
    "D1_DESC": "GROWING OF SOYBEANS FOR SELL",
    "D2_MAIN_ACTIVITIES": "GROWING AND SELLING OF SOYBEANS
     TO VENDORS IN KGS AT HOME",
    "ISCO_CODE_AI": "6111"
  }
]
```

Explanatory Notes: ISIC

The following explanations are provided to LLMs to guide reasoning on the ISIC classification task.

Listing 4: Few-Shot Prompt ISIC

```
"""
    Read the explanations first, then study the single JSON array.

    Why these ISIC labels are correct (human notes; not JSON):
    - ID 250 (Soldier): Code 8422 specifically refers to defence activities.
    - ID 742 (Chicken seller): This code is challenging because we do not know
        where the chicken is sold. However, as Zambia has a high informality rate,
        we can assume sale is from stalls and markets and assign 4781.
    - ID 256 (Kachasu brewer): Kachasu is a traditional distilled liquor which is
        homebrewed in Zambia. Because it is distilled, we should assign 1101.
    - ID 748 (Farm worker): Maize is a cereal and soybeans are oil seeds, so we
        assign 0111.
    - ID 846 (Farm labourer): Maize is a cereal, so we assign 0111.
    - ID 226 (Vegetable seller): This code is challenging because we do not know
        where the vegetables are sold. However, as Zambia has a high informality
        rate, we can assume sale is from stalls and markets and assign 4781.
    - ID 0 (Shift controller): Copper is a non-ferrous metal, so we assign 0729.
    - ID 31 (Charcoal): In Zambia, charcoal is commonly sold on the side of the
        road, not in stores, stalls or markets. So, we can assign 4799.
    - ID 687 (Fried cassava seller): Cassava is a starchy root vegetable widely
        cultivated in Zambia. This code is challenging because we do not know where
        the cassava is sold. However, as Zambia has a high informality rate, we can
        assume sale is from stalls and markets and assign 4781.
    - ID 475 (Subsistence farmer): Maize is a cereal and soybeans are oil seeds, so
        we assign 0111.
    - ID 913 (Farmer soybeans for sale): Soybeans are oil seeds, so we assign
        0111."""
]
```

Immediately after the explanatory notes, the following JSON array provides concrete examples formatted as input-output pairs. Examples are on the page below to fit to page.

Listing 5: Few-Shot ISIC Examples

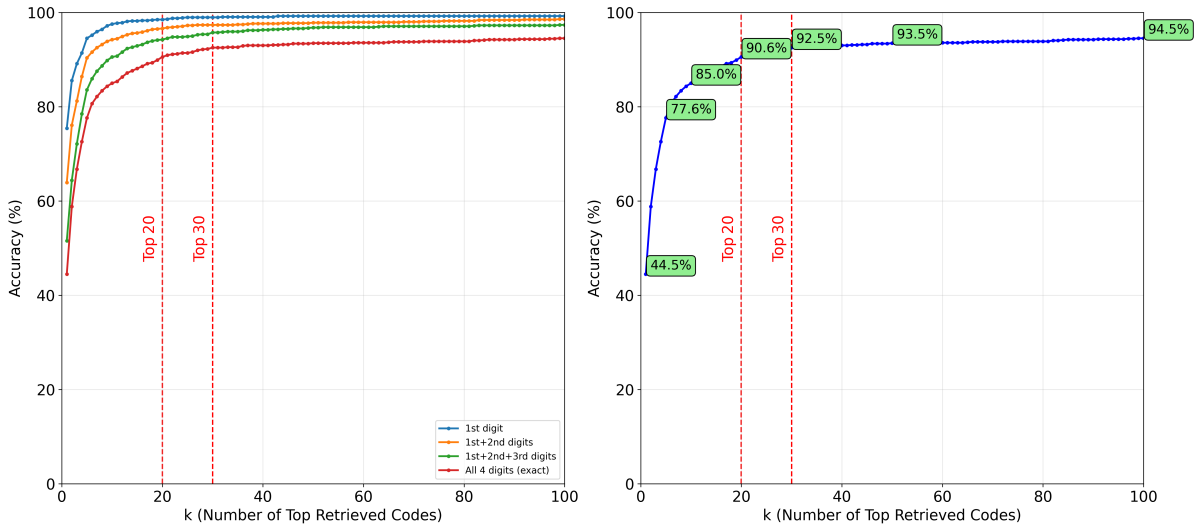
```
[
  {
    "D1_TITLE": "SOLDIER",
    "D1_DESC": "PROTECTING THE COUNTRY",
    "D2_MAIN_ACTIVITIES": "PROTECTING THE COUNTRY",
    "ISIC_CODE_AI": "8422"
  },
  {
    "D1_TITLE": "SELLING CHICKEN FOR SALE",
    "D1_DESC": "SELLING OF CHICKEN",
    "D2_MAIN_ACTIVITIES": "SELLING OF CHICKEN",
    "ISIC_CODE_AI": "4781"
  },
  {
    "D1_TITLE": "KACHASU BREWER",
    "D1_DESC": "SELLING KACHASU AT WHOLE SALE PRICE",
    "D2_MAIN_ACTIVITIES": "SELLING KACHASU BEER AT WHOLE PRICE",
    "ISIC_CODE_AI": "1101"
  },
  {
    "D1_TITLE": "FARM WORKER",
    "D1_DESC": "PLANT MAIZE AND SOYBEANS",
    "D2_MAIN_ACTIVITIES": "GROW AND SELL MAIZE AND SOYBEANS",
    "ISIC_CODE_AI": "0111"
  },
  {
    "D1_TITLE": "FARM LABOURER",
    "D1_DESC": "CULTIVATING LAND FOR PLANTING MAIZE",
    "D2_MAIN_ACTIVITIES": "GROWING OF MAIZE",
    "ISIC_CODE_AI": "0111"
  },
  {
    "D1_TITLE": "MARKETER",
    "D1_DESC": "BUYING AND SELLING OF VEGETABLES",
    "D2_MAIN_ACTIVITIES": "SELLING OF VEGETABLES",
    "ISIC_CODE_AI": "4781"
  },
  {
    "D1_TITLE": "SHIFT CONTROLLER",
    "D1_DESC": "SUPERVISING SMELTER WORKS",
    "D2_MAIN_ACTIVITIES": "MINING OF COPPER",
    "ISIC_CODE_AI": "0729"
  },
  {
    "D1_TITLE": "CHARCOAL BURNER/SALES MAN",
    "D1_DESC": "SELLING CHARCOAL",
    "D2_MAIN_ACTIVITIES": "SALE OF CHARCOAL",
    "ISIC_CODE_AI": "4799"
  },
  {
    "D1_TITLE": "BUSINESS LADY (RETAILER)",
    "D1_DESC": "BUYING AND SELLING OF FRIED CASSAVA",
    "D2_MAIN_ACTIVITIES": "BUYING AND SELLING OF FRIED CASSAVA",
    "ISIC_CODE_AI": "4781"
  },
  {
    "D1_TITLE": "SUBSISTENCE FARMER",
    "D1_DESC": "SELL MAIZE GROW AND SELL MAIZE SOYBEANS",
    "D2_MAIN_ACTIVITIES": "SELL MAIZE GROW AND SELL MAIZE SOYBEANS",
    "ISIC_CODE_AI": "0111"
  },
  {
    "D1_TITLE": "FARMER",
    "D1_DESC": "GROWING OF SOYBEANS FOR SELL",
    "D2_MAIN_ACTIVITIES": "GROWING AND SELLING OF SOYBEANS
     TO VENDORS IN KGS AT HOME",
    "ISIC_CODE_AI": "0111"
  }
]
```

C.4. RAG

Our method employs a Retrieval-Augmented Generation (RAG) system to improve accuracy and significantly reduce inference costs. We use the 3072-dimension `gemini-embedding-001` model from Google (Google AI for Developers, 2025). We encode documents with the task type set to `RETRIEVAL_DOCUMENT` and respondent data with the task type set to `RETRIEVAL_QUERY`. All codebook descriptions are encoded for ISCO and ISIC. At inference time, the 20 most relevant documents are retrieved and supplied to the LLM as candidate options alongside the most similar 'not elsewhere classified' code (see Section 3.3).

To benchmark the RAG system's effectiveness, we calculate the frequency at which the ground truth code is included within the selected codes. Figure 8 reports these frequencies with the number of retrieved codes ranging from 1 to 100. With 20 codes retrieved, the ground truth code is captured with 90.6% accuracy for ISCO and 88.1% accuracy for ISIC. From manual error analysis, most errors are caused by ambiguous respondent data or incorrect ground truth codes.

A. ISCO



B. ISIC

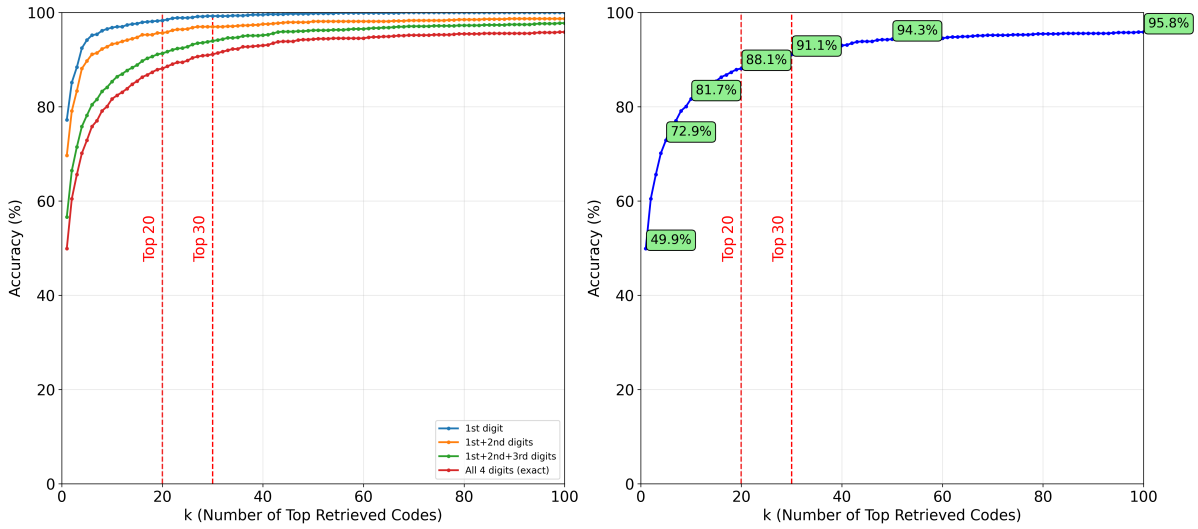


Figure 8: RAG Evaluation. Performance of retrieval-augmented generation (RAG) models across **A.** International Standard Classification of Occupations (ISCO) and **B.** International Standard Industrial Classification (ISIC) codes. Accuracy improves as the number of retrieved examples (k) increases, but decreases with higher code granularity (moving from fewer to more digits). At $k = 20$, the model achieves 90.6% accuracy on four-digit ISCO classification and 88.1% on four-digit ISIC classification.

D. Costs

Provider	Model	Total Cost ($)
OpenAI	GPT-4 Turbo	137.12
Google	Gemini 2.5 Pro	49.60
OpenAI	GPT-4.1	21.90
OpenAI	GPT-5	21.69
OpenAI	GPT-5 mini	3.60
Nebius AI Studio	Llama 3.1 70B	1.88
OpenAI	GPT-5 Nano	1.07

Table 8: Model Costs. This table shows the cost of classifying 1,000 observations. It includes all costs, including input tokens, output tokens, and thinking output tokens for reasoning models.

E. Technical Implementation

Before executing the classification code, users may configure a set of parameters that govern LLM behaviour. Parameter choice is influenced by the selected LLM's characteristics, including its ability to perform reasoning tasks, token and rate limits, and cost considerations. A full list of parameters is available in Table 9.

Parameter	Purpose	Default Value
API Provider	Python-LLM interface	OpenAI
Model	Desired LLM	GPT-5 Nano
Concurrency	Batch size to send to the API provider at once	20
Wait Time	Time between batches (seconds)	0.2
Temperature	Randomness of model output (0 = deterministic)	0
Max Tokens	Maximum number of tokens the model can use	30000
Seed	Random seed for generating few-shot prompt and initializing LLM client	42
RAG	RAG filtering parameter (0 = no RAG, >0 = top-k relevant codes from hybrid RAG search)	20
Sample	Sample size (testing)	None
Save Interval	Save progress every n chunks (0 = no periodic saving)	None
Partial Filename	Filename to save partial results	None
Examples	Few-shot examples included	0
Examples Dropped	IDs of removed few-shot examples	0

Table 9: Parameters. This table presents the set of parameters that can be configured by the user prior to executing the LFS classification code.

The classification pipeline is built around $i.$ an asynchronous LLM, with model endpoints available in Table 10; and $ii.$ a higher level processing routine that manages large-scale inference across survey responses. The LLM client provides a compatible interface through which prompts are submitted and model outputs are retrieved. The models are instructed to return JSON-formatted responses for consistency, which are cleaned and parsed in Python before being converted into structured tabular data. As LLMs may occasionally deviate from requirements or fail to produce a response, the client incorporates a retry mechanism with exponential backoff. Each request is attempted up to three times, with delays that double between retries, reducing the likelihood of repeat failures due to rate limits or transient network issues.

Provider	Model	API Endpoint
OpenAI	GPT-4 Turbo	gpt-4-turbo
Google	Gemini 2.5 Pro	models/gemini-2.5-pro
OpenAI	GPT-4.1	gpt-4.1
OpenAI	GPT-5	gpt-5
OpenAI	GPT-5 mini	gpt-5-mini
Nebius AI Studio	Llama 3.1 70B	meta/llama-3.1-70B
OpenAI	GPT-5 Nano	gpt-5-nano

Table 10: Model Endpoints. This table shows the different model endpoints that are used in our API provider client functions for the classifications.

The main processing function assigns ISCO and ISIC codes to all survey responses. When large datasets are provided, the function supports chunked execution by splitting the input into batches with a user-specified save interval. Failed or missing responses are replaced with placeholder entries, ensuring that no records are lost. For each chunk, ISCO and ISIC predictions are generated separately, optionally augmented with few-shot examples, and subsequently merged on their unique ID. Progress is monitored in memory for GUI use or written directly to the disk in notebook mode, allowing for recovery of partial results. In cases where chunks are not required, the function defaults to a single-pass pipeline that processes all items sequentially before merging results.

F. Practical Implementation

For the practical deployment of the LFS methodology within ZamStats, the team developed a graphical user interface (GUI) based on `Tkinter`. It is used in conjunction with `PyInstaller` to wrap all necessary Python packages and input files into one executable file.

How to open: The user must download the .exe file. The user will then be prompted to input valid API keys for relevant model providers (OpenAI, Anthropic, Google or Nebius AI Studio). The user may skip providers by leaving the API keys blank.

How to use: The user uploads an Excel spreadsheet with the specified input columns (see Section 3.1). To adjust the default settings, the user may specify the API provider, LLM, temperature, concurrency, maximum tokens, save interval, and optional few-shot prompting, few-shot examples, and RAG examples. The list of default settings is available in Table 9.

To run the application, the user must click 'upload file' followed by the 'run classifications' button. A progress bar at the bottom of the application shows the application's status and flags any errors that occur. After the classifications are completed, the file may be saved to the user's device.

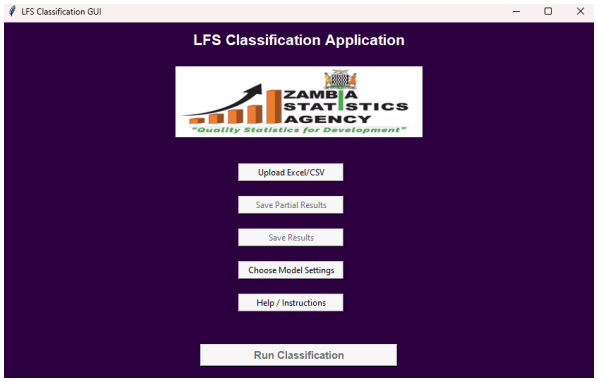


Figure 9: GUI. This is the .exe file interface used to upload files and run the LFS classification code.

theigc.org