Automated text coding: Census 2021

Automated coding methods for write-in text responses from the Census 2021 online questionnaire and their results.

Contact:
Census Customer Services
census.customerservices@ons.gov.uk
+44 1329 444972

Release date:
3 April 2023

Next release:
To be announced

Table of contents

1 . Main points

- Developing an in-house coding tool for Census 2021 was a new approach for the Office for National Statistics (ONS), but it had many advantages, including that the coding tools could be adapted.
- A second benefit was that automated coding enabled data to be available earlier than usual, speeding up analysis and updates to the tool.
- As the census electronic questionnaire was available approximately four weeks prior to Census Day, data were available to identify new terms that were not encountered before Census 2021, meaning that the coding indexes were up to date.
- Subject matter experts undertook real-time analysis of the most frequently occurring records that failed to code; this determined whether responses were suitable for inclusion in the coding indexes, and updated indexes could then be adopted into the pipeline for subsequent runs, ensuring the index was evolving in real time.
- Indexes were quality assured in real time; subject matter experts' recommendations were included as a standing item at the daily Census 2021 quality meetings, so the turnaround time from recommendation to implementation was efficient.
- Updating the indexes promptly meant that fewer records were sent externally for clerical resolution, ensuring timeliness and cost efficiency.
- Having the data early also allowed for quality assurance to be undertaken sooner; this meant that any issues were resolved before the end of the live-running phase and data were delivered to later stages without delay.

2 . Overview of automated coding

Census 2021 captured information about variables such as country of birth, ethnic group, occupation, and industry through text write-in boxes on the questionnaires. The process to code this written-in text compared written text against pre-defined indexes of terms. If the match was assessed by the matching method as sufficiently close to a term in the indexes, the process then assigned a numerical code. The coding strategy provided a consistent and standardised set of classifications on which to base statistical analyses and compare alternative datasets.

At the Office for National Statistics (ONS), we used the coding tools described in this methodology to code responses received from the online census questionnaires. Our questionnaire management supplier coded responses from the paper questionnaires.

Classifications and indexes

A classification essentially contains the categories that are supplied as an output to enable analysis. The following example shows "teaching and other educational professionals" in our Standard Occupational Classification (SOC) 2020.

2311 Higher education teaching professionals
2312 Further education teaching professionals
2313 Secondary education teaching professionals
2314 Primary education teaching professionals
2315 Nursery education teaching professionals
2316 Special and additional needs education teaching professionals
2317 Teachers of English as a foreign language
2319 Teaching professionals not elsewhere classified (n.e.c.)

These categories are populated by matching write-in responses to index entries. Index entries are lists populated with known responses used to group similar responses into the classification groups. For example, this is a sample of the index entries that populate the "higher education teaching professional" classification category:

2311 Lecturer, midwifery

2311 Lecturer, nursing

2311 Lecturer, political

2311 Lecturer, polytechnic

2311 Lecturer, university

Search-as-you-type functionality

To help collect write-in responses, the online census questionnaire presented people with a drop-down list of index suggestions matching the text that they were typing. People could either select one of the index options presented or continue writing their own response. The response they submitted would then progress through the automated coding processes. Search-as-you-type (SAYT) functionality was only available for the questions on:

- country of birth
- passports held
- ethnic group
- national identity
- main language
- religion

Read more about SAYT in our Search-as-you-type and address look-up functionality for Census 2021 article.

3 . Results and evaluation

Once automated and clerical coding was completed, the Census Quality Assurance team assessed its quality to identify if the key performance indicators (KPI) had been achieved. The Census Quality Assurance team analysed a 1% sample of coded records to assess the quality of the coding, and this is summarised in the following table.

Table 1: Key performance indicators for automated coding and achieved quality

Variable	% Automatically coded	Quality KPI %	Achieved quality %
Ethnic Group	94.7	97%	99.80%
National identity	97.2	98%	99.80%
Main Language	98.2	97%	99.90%
Religion	93.4	96%	99.90%
Country of birth	99.4	98%	~100%
Passports held	96.8	97%	99.30%
Sexual orientation	82.6	*	90.50%
Gender identity	86.9	*	94.00%
Occupation	65.3	89%	97.10%
Industry	52.7	88%	97.00%

Source: Census 2021 from the Office for National Statistics

Notes

1. No key performance indicator (KPI) was set for the new questions on sexual orientation and gender identity.

4 . Automated coding of socio-cultural variables

Parsing

Parsing refers to a series of methods used to enhance the chance of matches being made regardless of any syntactical or grammatical difference between index entries and write-in responses.

A common problem with write-in responses is that they can be prone to error, containing spelling mistakes or unexpected characters. There may also be new terms that have not been encountered previously. So, we carried out several pre-matching parsing steps to make it easier to code.

In this stage, we "cleaned" text strings using four parsing methods. We:

- removed white spaces from the front and end of records, and multiple white spaces between words to make input text format comparable with the index to help matching
- changed double letters to single letters, to help with spelling errors
- removed words that were not directly connected or useful, and which were unhelpful for automated coding, for example, removing the unnecessary text "I was born in" from "I was born in France" when answering about country of birth
- ensured that we did not apply the preceding three methods to words for which it would be inappropriate to do so

The original response always persists despite any parsing that we might apply to the write-in response for matching purposes.

Automated coding tool

Once we had completed the parsing steps, we moved responses to the matching stage of coding. We developed an automated coding tool for the socio-cultural variables in Census 2021. We then used it to code:

- country of birth
- passports held
- ethnic group
- national identity
- main language
- religion
- sexual orientation
- gender identity

The Office for National Statistics (ONS) Data Architecture team developed a separate coding tool to reflect the extra methods needed to code the variables of occupation and industry. This was because of the complex nature of coding these variables.

Matching method

The socio-cultural coding tool used Levenshtein distance to determine the closeness of the match between the write-in response and the socio-cultural topic indexes. The Levenshtein distance calculated the number of edits needed to transform the write-in text into the index entry. The fewer edits needed, the higher the confidence of a successful match. The Levenshtein distance produced a score between 0 and 1, with 0 meaning no match and 1 an exact match.

The balance for automated coding is to code as many records as possible while keeping the quality high. The socio-cultural coding tool selected a match once it reached the score threshold of 0.8. The Census Processing and Census Quality Assurance teams carried out quality testing using this score. They identified it coded a high proportion of records while ensuring that quality targets were achieved.

Process

The socio-cultural automated coding tool progressed write-in responses through several matching steps.

1. Automatic fails

Firstly, the tool compared records with an index of responses known to be not codable. These were phrases such as "don't know" or "not known". The tool then classified these matches as being "missing". The value of this strategy was that no further attempts were made to try and identify a code for these responses; this saved time and computational resource, and it prevented not-codable responses being sent for manual resolution.

2. Automatic exact matching

In this stage, the tool applied numerical codes to write-in text responses that matched exactly with expected text strings included in the index. We adopted this method because we could carry it out very quickly. It also ensures that fewer responses are sent for statistical matching using the Levenshtein distance, which requires more computational resource and so is a slower process.

3. Automatic statistical matching

Finally, the tool applied numerical codes to write-in text responses that closely matched to expected text strings included in the index with a high level of probability determined by the Levenshtein distance. This strategy allowed small respondent errors such as simple spelling mistakes to be coded correctly.

During the exact matching and statistical matching stages, the tool used two indexes for matching.

The first was the "replacement words" index, which replaced known synonyms for matching purposes. For example, "New Zealander" replaced a response of "Kiwi" for national identity.

The second index was the main index for the variable being coded.

Post-matching steps

Clerical resolution

Expert coders manually reviewed write-in responses that the automated coding strategies could not code. These coders then attempted to assign a valid classification code based on topic knowledge and other information on the questionnaire, and with the aid of additional reference materials.

Index updates

We also assessed records that did not code to see if they were suitable to be included in the indexes. This approach enabled new terminologies to be identified during live census processing. This reduced the number of records needing clerical resolution, as once a new write-in term was included, future occurrences would be automatically coded. So, clerical coders would only be presented with new terminology once, making the assessment of uncoded records more efficient.

The automated tools output the most frequently occurring uncoded write-in terms for each socio-cultural variable. Subject matter experts assessed these records for suitability for adding to a coding index. The possible outcomes of the subject matter experts' assessments were:

- term is a suitable new index entry
- term is synonymous with an existing index entry, and so is added to the replacement words index
- term is not related to the variable being collected and cannot be coded, and so is added to the not-codable index

As we added new terminology and index entries throughout the processing of the census, we carried out a final processing run of the data to ensure all records were coded against the final versions of all indexes. Table 2 shows how many new index entries we identified for each socio-cultural topic.

Table 2: The number of new index entries identified during Census 2021 live running

Topic	New index entries
Country of birth	6
Passports held	6
Religion	51
Ethnic group	72
National identity	9
Main language	7
Sexual orientation	21
Gender identity	18

Source: Census 2021 from the Office for National Statistics

5 . Coding of occupation and industry variables

The coding of the occupation and industry variables followed the same principle as socio-cultural coding. However, because of the additional complexity of coding, they needed additional methods and indexes, and different matching steps.

First, all the data passed through pre-processing before being coded to the Standard Industrial Classification (SIC) and then the Standard Occupational Classification (SOC) using the appropriate steps. The pre-processing included parsing steps such as:

- correcting some spelling errors
- truncating some words
- removing white spaces

Matching methods

Parsed data that remain in the coding tool process are then passed through the matching stage. The SIC and SOC knowledge bases used in the matching process are listed in this section. We used exact matching methods first, followed by fuzzy matching, and finally, a coding with dependencies stage for SOC only.

Exact matching

The tool checks for exact matches between the parsed data and the parsed index. When it finds an exact match, it applies the relevant SIC or SOC code from the index to the data. These responses are referred to as "exact matches".

Fuzzy matching

Responses remaining in the pipeline that were not coded through the exact matching stage move to the fuzzy matching stage, which consists of word matching and phrase matching.

Word matching

Index-specific weights are calculated for words appearing in the index. Weights indicate how often words appear in the index. The weights are calculated according to the following formula:

$$weight\left(word\right) = 1 - \frac{\log(number\ unique\ codes\ in\ index\ with\ descriptions\ containing\ the\ word)}{\log\left(number\ unique\ codes\ in\ the\ index\right)}$$

Scores are calculated for combinations of descriptions in the data and index as follows:

$$score_w = 10\left(\frac{a + 2b}{3}\right)$$

Where:

$$a = \frac{2\left(count\left(words\ in\ common\right)\right)}{count\left(words\ CD\ phrase\right) + count\left(words\ input\ phrase\right)}$$

$$b = \frac{2\Sigma\ weight\left(words\ in\ common\right)}{\Sigma\left[weight\left(words\ CD\ phrase\right) + weight\left(words\ input\ phrase\right)\right]}$$

And:

Count (words_in_common) is a count of the words in common between the input phrase and the Common Database (CD)9 phrase being evaluated.

Count (words_CD_phrase) is a count of the number of words in the dictionary phrase being compared.

Count (words_input_phrase) is a count of the number of words in the input phrase.

Weight (words_in_common) is the weight of each of the words in common between the input phrase and the CD phrase.

Weight (words_input_phrase) is the word weight of each of the words in the input phrase.

Weight (words_CD_phrase) is the word weight of each of the words in the CD phrase.

The resulting calculated score will range between 0 and 10, as an increasing function of a or b.
The index description with the highest score is matched with the description in the data.

Phrase matching

Descriptions in the parsed index and parsed data are sorted such that words are in alphabetical order. Levenshtein distances are calculated between descriptions in the parsed index and parsed data.

Scores are calculated as:

$$10\ \times\ \left(1\ -\ \frac{Levenshtein\ distance}{2\ \times\ the\ length\ of\ the\ parsed\ description\ in\ the\ data}\ \right)$$

The index description with the highest score is matched with the description in the data. Scores greater than a pre-set threshold were kept. These thresholds are discussed later in this release.

Matching steps for industry variable

Not-codable industry description matching

First, we compared records with an index of responses known to be not codable. These will be phrases such as "don't know" or "not known". Then, we classified these matches as being "missing". The value of this strategy is that no further attempts were made to try and identify a code for these responses, increasing the efficiency of the work.

Ambiguous matching

We carried out a direct match against an ambiguous knowledge base. An ambiguous knowledge base is one created after analysing responses from other surveys and contains entries needing more information than is available in the industry description.

Exact matching

We carried out matching to the:

- census-specific industry index – a collapsed version of the full framework, which is bespoke to the census
- industry caselaw emerging activities index – this contains entries identified since the last revision of the Standard Industrial Classification (SIC), and which have set precedent on how such records should be coded
- government name index – this matches to the establishment name and, if a government department is matched against it, a suitable code can be assigned
- SIC census index - an index created from previously coded census data
- SIC Annual Register Inquiry (ARI) knowledge base – a knowledge base composed of activity descriptions taken from Office for National Statistics (ONS) business survey returns (derived from the ARI_SIC list held on GCode)
- establishment name index – a list of establishments for which the SIC code is known, that is, schools, hospitals, or GP surgeries

Matching steps for occupation variable

Not-codable job title matching

We carried out direct matching on a not-codable knowledge base. Responses that matched at this stage were assigned a "not codable" marker, as they contained no useful occupation information and were not sent to clerical resolution.

Ambiguous job title matching

We carried out direct matching on an ambiguous knowledge base. Responses that matched at this stage needed additional information to code them, and so were routed straight to the dependency stage.

Primary matching to SOC knowledge bases

We matched parsed responses still in the process against the following SOC knowledge bases.

SOC2020_ashe_tuning knowledge base

The ONS Classifications team developed this knowledge base using phrases and misspellings of job titles from survey data that can be coded but are not suitable as an index entry. We checked whether the responses were a:

- direct match
- word match using threshold 10
- phrase match using threshold 9.75

SOC_welsh_without_dependencies

This knowledge base contains Welsh language SOC 2020 tuning data where codes can be uniquely assigned from job title alone or with industry code. We checked whether the responses were a direct match.

SOC2020_gcode_index

This is a reduced version of the published index with 28,054 entries, where:

- qualifiers have been removed
- some entries were added to aid automated coding
- occupational defaults were included
- 10,976 synthetic job titles were added
- the Additional Qualifier and/or Industry term has been added to the job title

We checked whether the responses were a:

- direct match
- word match using threshold 8
- phrase match using threshold 9.5

Job title with dependency matching

We passed responses that matched to ambiguous knowledge bases and responses that had still not been assigned a code through the dependency stage using additional information to help assign SOC. The job title component of this matching uses the direct, word, and phrase methods. We matched against the following knowledge bases.

SOC _dep_ind_tuning

This base contains job titles from the 2001 Census and 2009 Rehearsal that can be coded from job title and industry code. For example, "assistant store manager, 6100" codes to 1150 in SOC, and "assistance manager, 9200" codes to 1256 in SOC. We checked whether the responses were a:

- direct match
- word match using threshold 10
- phrase match using threshold 9.5

SOC_dep_ind

This base contains entries from the full modified index with an industry code. For example, "machine adjuster, 1300" codes to 5223 on SOC, and "machine adjuster, 1720" codes to 8131 on SOC. We checked whether the responses were a:

- direct match
- word match using threshold 8.5
- phrase match using threshold 9.5

SOC _dep_qual

This base contains index entries where there is a qualified or professional dependency, for example, "electrical engineer" where a qualification is held. We checked whether responses were a:

- direct match
- word match using threshold 10
- phrase match using threshold 9.75

Default job title matching

As a final stage where a match had not been possible at all preceding stages, the tool was able to assign a "default" or less-specific SOC code for particular job titles, for example, "nurse" and "secretary".

SOC_defaults

We checked responses against less-specific SOC codes as to whether they were a:

- direct match
- word match using threshold 10
- phrase match using threshold 9.75

The data output from the tool is:

- coded – assigned a valid SIC or SOC code
- residual – not coded and should be sent for clerical coding
- not codable – not coded and should not be sent for clerical coding

SIC and SOC index update summary

Over the period of live processing, we added 2,400 entries to the SIC knowledge bases and 850 to the SOC knowledge bases. These entries were added through the ONS Classifications team's weekly fails analysis work, as described in "Index updates" in Section 3: Automatic statistical matching.

6 . Related links

Delivering the Census 2021 digital service
Article | Released 4 October 2021
How the technical aspects of the Census 2021 digital service were built, for interest of digital professionals across government.

Quality and methodology information (QMI) for Census 2021
Methodology | Last revised 26 January 2023
Details the strengths, limitations, uses, users and methods used for Census 2021, England and Wales.

7 . Cite this methodology

Office for National Statistics (ONS), released 3 April 2023, ONS website, methodology, Automated text coding: Census 2021