New York City Directories Extracted Persons Entries, 1850-1890

Detailed Guide

Nicholas Wolf, orcid.org/0000-0001-5512-6151

Wesley Chioh, orcid.org/0000-0002-9742-7144

Stephen Balogh

Bert Spaan

Introduction

The files contain extracted entries from a series of New York City directories digitized by the New York Public Library (see https://digitalcollections.nypl.org/collections/new-york-city-directories) that listed the names, occupations or type of business performed by that individual, work address, and often the individual's home address or home town for thousands of residents of the city. Included directories are John Doggett's *New York City Directory* (1850 and 1851) and John Trow's *The Directory of the City of New-York* compiled by Henry Wilson (1852-1890). In later years the latter was known simply as *Trow's New York City Directory.*

Significance

While the directories present a snapshot of the city's life that is manifested through a commercial lens, unlike the separate business directories produced by the same publishers, which organized listings around sectors of the city's business life, the city directories place the emphasis on the individuals they chose to include. Thus, the directories were presented alphabetically by surname, along with occupation or business pursued by the individual, rather than business name. Moreover, home addresses were included in addition to business addresses, though in many, if not most cases, the single address listed encompassed both place of work and home. The inclusion of the label "colored" to refer to listings of the city's Black residents, and "widow of" with the name of a deceased husband to describe female residents also augments the cultural information available in the directories.

The directories have long been a wealth of information in their print format, but it is harder to grasp the larger picture of the city they depict because of the one-dimensional organizational principle in which they are presented. Organized alphabetically by name, it is easy for researchers to find a known individual they seek, including in digital editions such as those offered on the genealogical platform Ancestry.com where search by name is possible. But to pull the names of a single occupational category, for example, or to assemble the names of all individuals located on a single street requires scanning the entire directory to compile the relevant listings. By extracting all listings in computer-readable form, the present dataset enables multiple access points for filtering, grouping, and interpreting the full set of data for all directory years included.

Project Background

An earlier initiative to produce a mass extraction of entries from the directories was led by Bert Spaan of the New York Public Library Space/Time Directory project (http://spacetime.nypl.org), with the assistance of Stephen Balogh (New York University) and Nicholas Wolf (New York University). That project developed computational methods for text-column detection and field identification (separating text entries into labeled name, occupation, and address tokens), and was completed in November 2017. It focused on the 1849 to 1879 New York City directories.

Column detection and field labeling provided the primary difficulties in extracting entries from the directories, which would otherwise involve a straightforward parsing of the output generated by applying Optical Character Recognition (OCR) software to each page. The directory printers naturally sought to squeeze as many entries as possible onto a single page so as to create a more compact edition. Thus, from early in the nineteenth century as the number of entries grew, publishers arranged the entries in two columns; by century's end three-, four-, and, five-column layouts were deployed. This heightens the risk of OCR software generating an incorrect bounding box that combines an entry from the lefthand column with that on the right, treating it as a single entry. Narrowly spaced lines also encourages OCR software to incorrectly gather two consecutive entries together as a single multiline entry. The difficulty is compounded by entries that flow onto multiple lines, which the printers marked using indents for all subsequent lines after the first. Because identifying that indent depends on precise bounding of OCR coordinates around a line, any routine OCR errors quickly cause subsequent lines to be difficult to link back to head-line entries. Together, especially when combined with the usual character-level errors that even the best OCR software will produce, these features of the original text can lead to garbled outputs that must be fixed in pre- or post-processing steps. With approximately 8 million entries to be found in the forty years of directories selected here, fixing these entries in anything other than an automated fashion is daunting.

Even a perfectly captured entry poses a challenge for the other major task required to parse the information contained in the directories, that of identifying the borders between the tokens referring to the name of the individual, those referring to occupations, and those referring to addresses (and within address tokens, separating out the semantic borders between multiple addresses). Text markers like punctuation are not sufficient to identify these fields. For example, while commas were often placed between the three fields, the presence of a period (either to mark a name initial, or to show an abbreviation of an occupation word) lead to the omission of a period. And in any case, periods and commas are notoriously easy for OCR software to miss. Knowing an expected list of tokens to be found in each field is also insufficient. The token "smith" for example, may be appropriate as a name or an occupation. And any character-error in the OCR renders a token unmatchable to known lists.

The earlier project approached these problems using spatial distributions of lines on a page, as revealed by bounding box coordinates, to establish probable locations of column edges; a line offset from that edge was then appended to the next-closest line location. Spaan's script for producing this result (see https://github.com/nypl-spacetime/hocr-detect-columns) first clustered x1,y1 coordinates (the upper lefthand corner of each line's bounding box) to reveal the likely x-coordinate locations of column edges and indents based on multiple x-coordinate means. Once indent lines had been identified, it then searched through the other points to find the nearest point (identifying the parent line, to which the indented child line could be appended). The approach for identifying component parts of entries depended on a supervised machine learning technique, Conditional Random Fields (CRF). Balogh's script for parsing entries makes use of the Python sklearn-crfsuite module's implementation of CRF and a set of training data consisting of pre-labeled entries. Together, these two steps were able to produce a reasonably accurate set of records that were published on the Space/Time Directory website.

Present Project Extensions

The present continuation of this project aimed to expand beyond the initial directory years to take in the directories for 1850 to 1890. And it sought to improve accuracy as much as possible through preprocessing and postprocessing of the image and OCR-output files. Specifically, the following steps were taken:

- Images were deskewed (deskewing operations are built into the OCR software used, Tesseract version 4, but it was felt a strong deskewing might assist in improving quality).
- Grayscale versions of the page images were temporarily dilated to enable identification of the margins of the entries as a text block. The pages were then cropped at those locations with the goal of eliminating marginal markings, headers, and advertisements as much as possible.
- Pages with full-page advertisements were removed by hand to avoid trying to account for their elimination in the OCR output parsing stage.
- The cropped grayscale jpegs were given a fresh OCR run using Tesseract 4.1, resulting in HOCR output files.
- Additional training data were generated for the CRF labeling scripts.
- When parsing the HOCR files, bounding boxes for lines were assessed to identify those falling within the bounds of the text block representing the entries. Text whose bounding box coordinates fell outside of those limits were ignored.
- When identifying indented lines, a new approach was deployed whereby lines were first sorted into vertical-axis order (y-axis) based on bounding-box coordinates. This enabled identified indent lines to be appended to the previous line based on entry order, rather than appending them to the nearest line based on Cartesian distance. This helped avoid errors caused by irregularly drawn bounding boxes that had led to the closest-distance line being one other than the preceding line.
- Vertical placement of captured lines on the page were assessed to identify gaps in captured lines, thus enabling identification of pages in which the OCR software had skipped parts of the text.
- Occupation and address tokens were assessed for frequency across the entire corpus. Tokens occurring with high frequency were checked for accuracy and confirmed to be correct. Less-frequently occurring tokens were matched using Levenshtein distance to confirmed tokens, then checked as to whether they were OCR errors or abbreviations. If errors, the incorrect token was swapped with the established high-frequency correct tokens. If an abbreviation, the abbreviated token was swapped out with an established unabbreviated token.
- All occupations and addresses were given a score of likelihood of correctness. Those scoring highest (15) had been confirmed by hand to be correct (or corrected by a swap-in of the correct/unabbreviated token). Those scoring lowest (1) were deemed to be very likely incorrect, having been unable to find a proper swap for the token and given the presence of the token only once across the entire corpus. Intermediate scores (2-14) were given to tokens for which likelihood of correctness could not be determined, but for which frequency of occurrence (between 2 and 14 times in the corpus) suggests that they may not have an error. See the "Correcting and Scoring" section below for more details.
- Flags were attached to the final data records to indicate where the terms "colored" or "widow" had been used to describe the person in an entry.

Data Structure

The data are presented in NDJSON (i.e., newline-delimited JSON format, see https://github.com/ndjson/ndjson-spec), UTF-8, Unix LF encoded files. As per the NDJSON specifications, each line consists of a valid JSON object (see RFC 7159, https://www.ietf.org/rfc/rfc7159.txt). This format provides for easier parsing based on newline separation without losing the affordances of JSON-formatted data.

Each record (i.e. line) corresponds to a single entry in a city directory, which in turn records the information about a single person in that directory's year. Each record is fully self-describing, including all information about the directory's NYPL identifier, directory year, page identifier, line number, unique entry identifier, location on the page, and entry contents.

For ease of use, the entries have been kept separated into directory year, so that each files corresponds to a single directory. The naming convention of each file is:

YEAR.NYPL_DIRECTORY_UUID.ndjson

e.g., for the 1850-51 directory,

1850.4adf9ec0-317a-0134-03ad-00505686a51c.ndjson

Variables

name	description	jsonType	universeValues	encodedMeaning	required

name	description	jsonType	universeValues	encodedMeaning	required
directory_uuid	The universal unique identifier (UUID) assigned by the NYPL to each directory. This ID can be found on the record page for each directory in the NYPL Digital Library and used to access the same directory over the NYPL Digital Library API.	string	n/a	n/a	yes
page_uuid	The UUID assigned by the NYPL to each image file, corresponding to one page of a directory, prepended by the sequential image number (one for each image that makes up a full directory). Note that directories typically start their entries around 20 pages into the volume. Moreover, within the pages of entries are full-page ads that have been excluded as they do not contain entries. Thus, image numbers should be expected to start on a number other than 1 and include some gaps.	string	n/a	n/a	yes
entry_uuid	A UUID assigned in the course of the data extraction to uniquely identify each entry.	string	n/a	n/a	yes
original_hocr_line_number	Allows matching of entry back to the original HOCR line number of the OCRd file for each page. In the HOCR generated by Tesseract, each identified line, consisting of token-by-token bounding boxes, is assigned its own bounding box coordinates and a unique line number as an HTML attribute on the element corresponding to that line. This ID corresponds to the final number of that line number attribute, e.g. for "id=line_1_2" the original HOCR line number is 2.	string	whole numbers 1 to approximately 200	n/a	yes

name	description	jsonType	universeValues	encodedMeaning	required
bbox	The x1, y1, x2, y2 bounding box coordinates, in that order and measured in pixels, of the OCRd line. HOCR bounding box coordinates are oriented to a 0,0 origin point at the top lefthand side of the page. The top lefthand coordinate of the bounding box will be x1, y1. The x1 is the distance from the lefthand side of the page, and y1 is the distance from the top of the page. For further reference, see http://kba.cloud/hocr-spec/1.2/#bbox.	string	n/a	n/a	yes
col	Column in which the entry was located based on the bounding box location of the line on the page. All directories in this 1850-1890 span had two-column layouts.	string	1 or 2	Column 1 or Column 2	yes
appended	True/false assertion about whether the entry included indented lines after the main headline that have been appended.	string	0 or 1	0 (false/no) or 1 (true/yes)	yes
skipped_line_after	Indicates whether an OCR error consisting of a skipped line following the current line occurred; if so this could mean that a subsequent indent line was present and was therefore not appended.	string	0 or 1	0 (false/no) or 1 (true/yes)	yes
directory_year	The year span listed on the title page of the directory for years covered.	string	1850-51, 1851-52 ... or 1889-90	n/a	yes
nypl_url	The permanent URL of the directory in the NYPL Digital Library.	string	n/a	n/a	yes
total_lines_directory_from_hocr	Number of lines detected by OCR on the page on which the entry exists. Can be used to detect pages for which OCR failed to detect all text on a page.	string	n/a	n/a	yes

name	description	jsonType	universeValues	encodedMeaning	required
complete_entry	The original full entry as detected by the OCR software, with minimal normalization (smart quotes were replaced with straight quote marks, tab characters replaced with single spaces, spacing was normalized around hyphens to assist with multiline hyphenation appending of indent lines).	string	n/a	n/a	yes
labeled_entry	A JSON array consisting of a series of JSON objects generated out of automated labeling of component parts of the complete entry using Conditional Random Fields (CRF).	array of objects	n/a	n/a	yes; must include at least one labeled entry component (will not be an empty array)
labeled_entry.subjects	Components of the original entry labeled as a subject, i.e. a name of a person.	array of strings	n/a	n/a	yes, but may be an empty array if no subjects found owing to OCR or labeling error.
labeled_entry.occupations	Components of original entry labeled occupations for the entry.	array of strings	n/a	n/a	yes, but may be an empty array owing to OCR or labeling error.
labeled_entry.locations	Components of original entry labeled address or spatial locations.	array of objects	n/a	n/a	yes, but may be an empty array owing to OCR or labeling error.
labeled_entry.locations.value	An address or location included in the entry.	string	n/a	n/a	yes; if there is a location, every location will have at least a value.
labeled_entry.locations.label	Additional refining information attached to a address or location such as "rear," "home," or "foot".	string	ft, r, h, or variations of these values	ft (foot), r (rear), h (home)	no
corrected_entry	A modified version of the labeled_entry. A JSON array consisting of a series of JSON objects listing subjects, occupations, and addresses for which proposed corrected versions have been created through postprocessing of the text.	array of objects	n/a	n/a	yes
corrected_entry.subjects	Components of the original entry labeled as subjects. No changes or postprocessing has been done on the subjects; the result will be the same as the labeled_entry value.	array of strings	n/a	n/a	yes, but may be empty array owing to OCR or labeling error

name	description	jsonType	universeValues	encodedMeaning	required
corrected_entry.occupations	Components of original entry labeled an occupation. These values have been postprocessed and swaps made token-by-token to replace values likely to contain errors with those considered correct; a score is attached to the resulting occupation to show likelihood of correctedness.	array of objects	n/a	n/a	yes, but may be empty array owing to OCR or labeling error.
corrected_entry.occupations.value	An occupation or business name attached to an entry's person.	string	n/a	n/a	yes, if corrected_entry.occupations is not an empty array.td>
corrected_entry.occupations.score	A score between 1 and 15 (see description of scoring below) showing the likelihood of correctedness of the occupation value. If a multi-token occupation, the score reflects the lowest of all scores across all tokens, unless marked a "widow" or "colored" individual, in which case score will be 15.	string	numbers 1-15	1 is least likely to be correct, 15 the most likely	yes, if an occupation is present
corrected_entry.locations	Components of the original entry labeled a address or location. These values have been postprocessed and swaps made token-by-token to replace values likely to contain errors with those considered correct; a score is attached to the resulting occupation to show likelihood of correctedness.	array of objects	n/a	n/a	yes, but may be empty array owing to OCR or labeling error
corrected_entry.locations.value	An address or location name attached to an entry, with tokens considered to be errors swapped with corrected versions.	string	n/a	n/a	yes, if the correct_entry.locations is not an empty array
corrected_entry.locations.label	Additional refining information attached to an address. These have been postprocessed to remove OCR errors and the universe of available value restrained to just three: rear, home, foot.	string	r, h, ft	ft (foot), r (rear), h (home)	no

name	description	jsonType	universeValues	encodedMeaning	required
corrected_entry.locations.score	A score between 1 and 15 (see description of scoring below) showing the likelihood of correctedness of the location value. If a multi-token location, the score reflects the lowest of all scores across all tokens.	string	numbers 1-15	1 is least likely to be correct, 15 the most likely	yes, if a location is present
labeled_black	True/false assertion about the presence of the label "colored" by the directory to describe an entry's person using "col'd", "colored", or "col". In the depunctuated version of the occupation, these will often appear as "cold".	string	0 or 1	0 indicates no label "colored" present, 1 indicates labeled "colored" presen	yes
labeled_widow	True/false assertion about the presence of the label "widow" by the directory to describe an entry's person using "wid" or "widow".	string	0 or 1	0 indicates no label "widow" present, 1 indicates labeled "widow" present	yes
low_entry_caution	True/false assertion about whether the number of entries extracted from the page are below one standard deviation from the average number of entries found per page in the particular directory. Indication that OCR missed enough entries to potentially lead to errors in the entries extracted.	string	0 or 1	0 false, sufficient entries, 1 true, insufficient entries	yes

Correcting and Scoring

Occupation and address/location tokens were corrected using a system whereby all tokens were clustered using their bi-gram fingerprint representation. This fingerprint method depunctuates the token string, then sorts its unique two-character combinations alphabetically. The result is a strongly unique representation of a token's essential features. Clustering using this methods brings together highly similar tokens separated only by the presence of typical OCR errors such as additional punctuation characters. This step can be thought of as a dimensional reduction of the data so that only a subset of the overall tokens need to be addressed and the correction performed there perpetuated to the rest of the records.

The most common token from among these clusters was identified. Then, if the clusters were sufficiently large, they were examined by hand to ensure that the most common token value selected was a correct token. The corrected tokens were then swapped in for the incorrect tokens and the whole corpus re-clustered.

In this second round of clustering, a looser, Levenshtein distance matching was conducted, matching low-frequency (small-size cluster values, and therefore more likely to be errors) against high-frequencye/large cluster token values. Once again, frequently matched connections were hand-checked to ensure that a correct swap of tokens had been identified.

Once these correction swaps were assembled, and combined with as many swap-outs for token abbreviations used in the directories as could be identified (so that abbreviations would not be confused with an error in the token), the entire corpus was clustered once again to determine how often any given occupation token or full address (including address number) occurred in the full corpus. This is the basis of the score attached to each occupation or location.

A score of 1 indicates that the highest frequency any given token in the occupation occurs is once across the entire corpus of directories, suggesting strongly that the token has an error. For addresses/locations, a score of 1 indicates that the address only occurs once, again likely an error. For both categories, a score of 1 was also assigned to tokens for which a correction swap could not be made.

Conversely, an occupation or address scoring higher occurs much more often and can be more confidently considered correct.

A score of 15 means not only that the occupation or address is very frequent (15 or more, sometimes hundreds of times), but that it has been checked by hand to be confirmed as a valid address, meaning that no tokens within the address appear to be wrong. In practice, most tokens scoring above 5 or 6 will be found to be error free.

Counts and Features of the Data

The files consist of:

Entries

Total entries: 7,926,161

Total entries with all occupations and locations given score of 15: 5,780,451

Number of entries labeled "colored": 10,277

Number of entries labeled "widow": 651,158

Locations

Total locations (many entries contain more than one location): 10,000,895

Total locations with a score of 15: 7,791,916

Occupations

Total occupations (some entries lack an occupation because of OCR error): 7,860,612

Total occupations with a score of 15: 7,574,191

Example Single-Line JSON Record


```
{
  "directory_uuid": "4adf9ec0-317a-0134-03ad-00505686a51c",
  "page_uuid": "100.56750563.848ac750-5293-0134-73af-00505686a51c",
  "entry_uuid": "25e8262c62e211ea902028f076102196",
  "original_hocr_line_number": "2",
  "bbox": "94 83 872 144",
  "col": "1",
  "appended": "1",
  "skipped_line_after": "0",
  "complete_entry": "Chandler Job . Foster, varieties, 81 Maiden lane, h. 81 Avy.",
  "labeled_entry": {
    "subjects": [
      "Chandler Job"
    ],
    "occupations": [
      "Foster",
      "varieties"
    ],
    "locations": [
      {
        "value": "81 Maiden lane"
      },
      {
        "value": "81 Avy.",
        "labels": [
          "h"
        ]
      }
    ]
  },
  "directory_year": "1850-51",
  "nypl_url": "https://digitalcollections.nypl.org/items/7b3fbb00-5293-0134-b386-00505686a51c",
  "total_lines_directory_from_hocr": "173",
  "corrected_entry": {
    "subjects": [
      "Chandler Job"
    ],
    "occupations": [
      {
        "value": "Foster",
        "score": "1"
      },
      {
        "value": "varieties",
        "score": "15"
      }
    ],
    "locations": [
      {
        "value": "81 Maiden Lane",
        "score": "8"
      },
      {
```

```
        "value": "81 Avenue",
        "score": "1",
        "labels": [
          "h"
        ]
      }
    ]
  },
  "labeled_black": "0",
  "labeled_widow": "0",
  "low_entry_caution": "1"
}
```


Acknowledgements

This work was completed with assistance from a grant from the New York University Center for Faculty Advancement Curricular Development Challenge Fund.

License

Citation

Nicholas Wolf, Wesley Chioh, Stephen Balogh, and Bert Spaan, "New York City Directories Extracted Persons Entries, 1850-1890," New York University Faculty Digital Archive, 2020, http://hdl.handle.net/2451/61521.