Journal of
open research software

Osborne et al 2014 Historical Post Office Directory Parser (POD Parser) Software From the AddressingHistory Project. *Journal of Open Research Software,* 2: e23, DOI: http://dx.doi.org/10.5334/jors.aq

SOFTWARE METAPAPER

Historical Post Office Directory Parser (POD Parser) Software From the AddressingHistory Project

Nicola Osborne[1], George Hamilton[2] and Stuart Macdonald[3]

[1] Author of paper, AddressingHistory Project Officer, Social Media Officer, EDINA, UK

[2] Co-author of paper, Developer of the POD Parser software, Software Engineer, EDINA, UK

[3] Co-author of paper, AddressingHistory Project Manager, Associate Data Librarian, EDINA, UK

The POD Parser is Python software for parsing the OCR'd (optical character recognised) text of digitised historical Scottish Post Office Directories (PODs) to produce a consistent structured format for the data and for geocoding each address. The software was developed as part of the AddressingHistory project which sought to combine digitised historic directories with digitised and georeferenced historic maps.

The software has potential for reuse in multiple research contexts where historical post office directory data is relevant, and is therefore particularly of use in historical research into social, economic or demographic trends. The POD Parser is currently designed for use with Scottish directories but is extensible, perhaps with some adaptation, to use with other similarly formatted materials such as the English Trade Directories.

Keywords: historical post office directories; text parsing; OCR; geocoding; Python; digital humanities; parser; Scottish history; post office directories

Funding Statement: Initial development of the POD Parser was funded by JISC as part of the Developing Community Content Programme [14]. Additional development of the current version of the POD Parser was funded internally by EDINA.

(1) Overview

Introduction

The POD Parser software was developed as part of the JISC-funded AddressingHistory project[1] (Apr. – Sept. 2010) to develop a community engagement web tool and Application Programming Interface (API) to enhance and combine data from digitised historical Scottish Post Office Directories(PODs) with contemporaneous historical maps.

TPODs emerged during the late seventeenth century to meet the demand for accurate information about trade and industry due to the expansion of commerce during this period. They offer a wealth of detailed information regarding residential names, occupations and addresses and as such are a fitting resource for both genealogical study and for understanding social, economic and demographic trends and changes within Scotland.

At the time of writing AddressingHistory focuses on Edinburgh, Glasgow and Aberdeen mapping and 9 Post Office Directories from late 18th to early 20th centuries. Both the website[2] and API[3] are currently used by academic researchers looking at the economic and social history of Edinburgh, with the API also incorporated into the Arts and Humanities Research Council's Visualising Urban Geographies project[4].

The website and API were developed by EDINA at the University of Edinburgh, in partnership with the National Library of Scotland (NLS), using materials digitised using Optical Character Recognition (OCR) techniques, stored as XML, and published as part of an on-going NLS and Internet Archive programme.

The POD Parser aims to parse the XML records and determine forename, surname, occupation and address(es) of each entry. Furthermore, each address location is geocoded using the Google Geocoding API [5].

The PODs contain both personal and professional address listings. They, also include miscellanea such as shipping information, professional body memberships, listings by profession, and adverts. For the project only the General Directory section of the directories – a listing of individuals and their workplace addresses - were parsed.

Currently over 750 PODs have been digitised as part of the NLS programme, all of which are available via the Internet Archive in the public domain [6]. The wide range of data collection practices, publishers, publication dates, and locations covered give rise to highly heterogeneous directories. The POD Parser is flexible and adaptable to variants in the General Directory format however customisation may be required when using the Parser with PODs of contrasting format.

The POD Parser code is open source allowing the Parser to be adaptable for parsing different directories within the PODs (e.g. Street Directory) or similar historical directories from other localities such as the English Trade Directories. This would, however, necessitate significant Parser re-configuration and customisation for each new style of POD or directory.

Implementation/architecture

The POD Parser is a platform independent command-line tool and library for parsing Scottish Post Office directories. The python application parses the directories from XML, and through a variety of string replaces, stop-words, address lookups and line return fixes attempts to repair OCR errors to create valid POD entries.

The podparser is made up of a number of classes for executing a parse run, modelling the structure of the POD, cleaning POD entries, geo-encoding entry addresses and storing results in the database.

The entry point of a run of the parser is the Parser class, an instance of which creates a Directory object that stores POD metadata and a list of pages (Page) to be parsed, each of which contains a list of entries (Entry) to be parsed. An instance of EntryChecker checks the structure of an Entry to identify the name, profession and addresses of the entry, making on the fly corrections to OCR problems. For each address that is identified, an instance of Google or GooglePremium will fetch the co-ordinates of the address. The google encoder can be executed independently of the parser, see [7].

If database details are specified, an instance of PodConnection will store entries in the database. Associated schema can be found in the code repository [8].

Input

The parser is designed to accept input files in the format and file structure of the Scottish Post Office directories djvu XML files. The parent directory should contain a metadata XML file ending in _meta.xml containing the following values:

```
<metadata>
  <volume></volume>
  <publisher></publisher>
</metadata>
```

The POD pages are expected in a child directory whose name ends in _djvu_xml.Each file contains a single POD page whose page number is contained in the file name.

If the POD page files required by the parser are not available in a child directory, they can be generated using the "podfetch" script [9]:

```
$ cd </path/to/pod>
$ podfetch -d <url>
```

If successful, this will fetch a metadata file and a djvu file containing all pages in the pod. A new djvu XML file is then generated for each page in the pod in a new directory. Please note that on slower internet connections this process can take a long time.

Example input files are available in the code repository [10] structured as shown in **Figure 1.** Further information on Input can be found in the pod parser documentation [11].

```
<LINE>Auld, John, grocer and victualler, 25 Duke street ;</LINE>
<LINE>house, 4 Burrell's lane.</LINE>
<LINE>Auld, John, painter and paperhanger, S9 Bath street.</LINE>
<LINE>Auld, John (of David Auld &amp;amp; Sons), house, 13</LINE>
```

Fig 1: Example input data.

```
 | Auld | John | grocer and victualler | G | 25 Duke street ;
house, 4 Burrell's lane.
> | 25 Duke Street, Glasgow, Scotland | 55.860184 : -4.238552 |
RANGE_INTERPOLATED | derived (Duke St)
> | 4 Burrell's Lane, Glasgow, Scotland | 55.860516 : -4.238328 |
GEOMETRIC_CENTER | derived (Burrell's Ln)

 | Auld | John | painter and paperhanger | F | S9 Bath street.
> | 9 Bath Street, Glasgow, Scotland | 55.863495 : -4.253631 |
RANGE_INTERPOLATED | derived (Bath St)
```

Fig 2: Example output data.

Parsing process and output

The parser can be used as a command-line application or invoked as a library call within a python script. The command-line application parses the Post Offices directories from XML and optionally commits the entries to a database. Used in either way the parser processes each file on a line-by-line basis.

Post Office directories can contain many pages, leading to parse times of many hours. In cases where many pages are being parsed it makes more sense to use a callback to process the results after the parsing of each page. This means if the process is killed before finishing, it can be restarted from the point of failure.

Each cleaned entry is geo-encoded using Google's geocoding api[5] and the results are printed to "standard out" as each entry is processed (see **Figure 2**).

Quality control

A variety of unit tests are provided to test database queries, google geocoding API connectivity and responses, specific OCR errors and general API code coverage. Integration tests are provided to validate database connectivity and SQL queries where appropriate.

The full range of available unit and integration tests are detailed in the PODParser code repository on Github [12].

The proportion of parsed records with a low accuracy of geo tag (as defined by receiving a Google geocoding "accuracy" score of less than 5), or the proportion of records with no geo tags after parsing, can act as a representative measure of Parser accuracy. These measures were used in the development of the POD Parser when making changes to accommodate new directories with variations in format or quality of POD entries.

The POD Parser's second round of development was also informed by a small project in which two postgraduate history students examined and documented the quality of output data, providing analysis of common issues and the accuracy of the POD Parser.

For instance the most accurately parsed POD currently available via the AddressingHistory website, Aberdeen 1881, has a geotag accuracy of 99% (percentage of Google geocoding with an accuracy of 5 or more). By contrast, the least accurately parsed POD currently available, Aberdeen

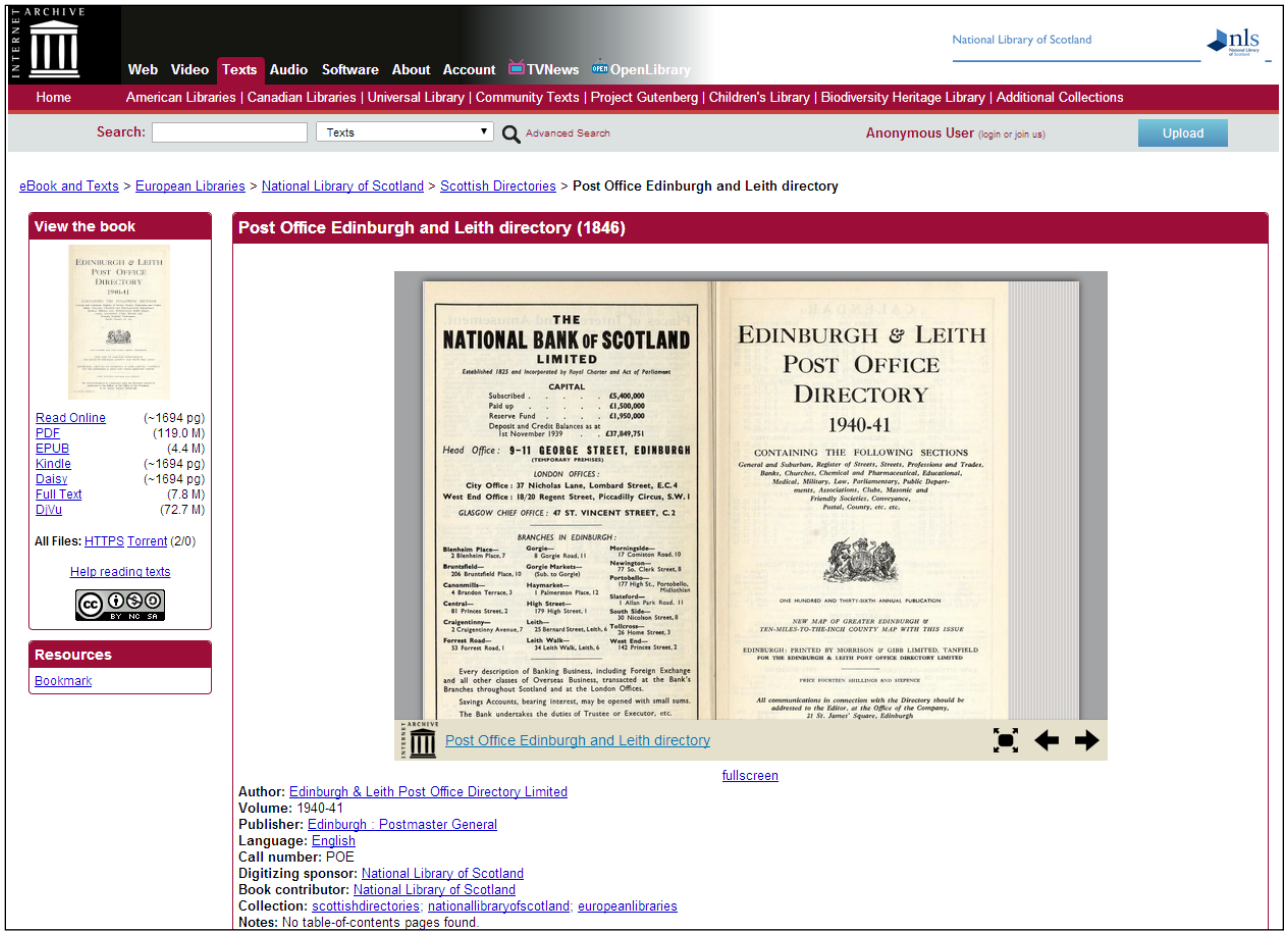

Fig 3: Screen capture of the Internet Archive page for the 1940-41 Edinburgh and Leith Post Office Directory. The red box on the left hand side of the screen indicates the location of the link to the All Files: HTTP area.

1891, has a geotag accuracy of 87%. The majority of the PODs parsed to date have an accuracy over 90% as a result of iterative rounds of testing and improvement to the POD Parser.

User contributions and feedback on the accuracy and issues encountered in output data (surfaced within the AddressingHistory website) also provide a form of ongoing quality assurance to inform future development of the Parser.

(2) Availability

Operating system

Platform independent.

Programming language

python2

Additional system requirements

An internet connection is required as part of the geo-encoding process. There are no other specific requirements. The requirements do, however, depend on the size of data set – the POD – being parsed. The database used for output must, therefore, have sufficient capacity to accommodate the parsed input data. There are two alternative methods of running the POD Parser that place different demands on the system with the page by page method more suitable for the Parser on large data sets. For more information see the "Usage" section in [11].

Dependencies

Python libraries; argparse and psycopg2 (latter only where Parser results are to be stored in a database – currently only Postgis is supported).

Google Geocoding API. The Parser requires use of a geocoding tool and uses the Google Geocoding API at present although it has been designed to be extensible to, e.g. Yahoo! BOSS Geo Services.

List of contributors

George Hamilton, Software Engineer at EDINA developed the current POD Parser (version 0.4) making significant developments and adaptations to the Parser. This work built upon the first version of the POD Parser (in 2010), developed by Joe Vernon, then a Software Engineer at EDINA.

Archive

Name

PyPI

Persistent identifier

http://pypi.python.org/pypi/podparser

License
GPL (General Public License) Version 3

Publisher
George Hamilton

Date published
07/01/2014 (v. 0.4)

Code Repository
Name
GitHub

Identifier
https://github.com/edina/podparser

License
GPL (General Public License) Version 3

Date published
27/05/2011 (v. 0.1)

Language
Git (repository); Python (Parser); SQL, Postgres/PostGIS (database); XML (configuration files); html, text (documentation).

(3) Reuse potential

The software has potential for reuse in extending the temporal and geospatial range of data available for existing research contexts (e.g. economic and social history).

The current collection of over 750 Scottish PODs are publicly available via the Internet Archive [6]. The XML files which the POD Parser uses as input are provided in the "All Files: HTTPS" area (see **Figure 3**) with the naming convention: postofficeann<YearName>_scandata.xml. Where Year is the year of the POD (e.g. 1888), and Name is an abbreviated form of the name of the POD which may reflect the author, or the area, covered by the POD (e.g. "peac" for "Peace's Orkney almanac and county directory").

The POD Parser also has the potential for use across multiple research contexts where historical post office directory data may be relevant either on it's own, or when combined with additional sources of data. For instance, the POD data may be used in research into historical health and epidemiology, town planning and architecture, and - as the PODs represents an unusual representation of women's lives and occupations - into the lives and roles of women.

The POD Parser is currently designed for use with Scottish directories, and for processing a particular format of file used for the PODS, but is extensible, with some adaptation, to use with other similarly formatted materials such as the English Trade Directories. The existing POD Parser could also be adapted to not only parse POD data but also combine each entry with complimentary data sources. The Parser could also be made more flexible, allowing the user to define the order, or enabling the Parser to accept alternative structured data.

Support for the Pod Parser software is available through the GitHut issue tracker (available within the code repository) or through contacting the authors of this paper. Additionally support for the Pod Parser, the AddressingHistory website and API is available via a form on the AddressingHistory website [13], or via the EDINA helpdesk (edina@ed.ac.uk).

Acknowledgements
In addition to the authors of this paper we would like to acknowledge the work of Ian Fieldhouse, Software Engineer at EDINA, who has worked on development of the AddressingHistory web tool which the POD Parser supports. We would also like to acknowledge the advice and support provided by members of the AddressingHistory Steering Committee particularly those from our partner organisations: Professor Richard Rodger of Edinburgh University, Chris Fleet of the National Library of Scotland Maps Library and Ines Byrne, Digital Project Officer at the National Library of Scotland.

Special thanks are extended both to Ines and to Lee Hibberd, Digital Preservation Officer at the National Library of Scotland, for making available the NLS stylesheet for the PODS, which has enabled us to add a new script to the PodParser improving its capacity for reuse with other digitised PODs. This improvement was influenced by the feedback of the JORS reviewers who we would like to thank for their comments and suggestions.

We would also like to thank professional genealogist Chris Paton, and AddressingHistory users and volunteer testers all of whom have provided feedback on the project website that have helped us to improve the usefulness and accuracy of the POD Parser.

References

1. **EDINA** 2013 AddressingHistory Phase 2. Available at http://edina.ac.uk/projects/addressinghistory2_summary.html [last accessed 20 February 2013].
2. **EDINA** 2013 AddressingHistory. Available at http://addressinghistory.edina.ac.uk/ [last accessed 20 February 2013].
3. **EDINA** 2013 AddressingHistory: Using the API. Available at http://addressinghistory.edina.ac.uk/api [last accessed 20 February 2013].
4. **National Library of Scotland** 2009 Visualising Urban Geographies. Available at http://geo.nls.uk/urbhist/ [last accessed 20 February 2013].
5. **Google** 2013 The Google Geocoding API In Google Maps API Web Services. Available at https://developers.google.com/maps/documentation/geocoding/ [last accessed 20 February 2013].
6. **National Libraries of Scotland and the Internet Archive** 2013 Internet Archive Scottish Directories (collection). Available at http://archive.org/details/scottishdirectories [last accessed 20 February 2013].

7. **Hamilton, G** 2012 Encoder.py. In Github: podpaser/geo. Available at https://github.com/edina/podparser/blob/master/podparser/geo/encoder.py [last accessed 9 June 2014].
8. **Hamilton, G** 2012 Schema.sql. In Github: podpaser/schema. Available at https://github.com/edina/podparser/blob/master/podparser/etc/schema.sql [last accessed 9 June 2014].
9. **Hamilton, G** 2013 POD set up. In Post Office Directory Parser (podParser). PyPI podParser v0.3 package documentation. Available at https://pythonhosted.org/podparser/#pod-set-up [last accessed 19 February 2014].
10. **Hamilton, G** 2012 example In Github: podpaser/test. Available at https://github.com/edina/podparser/tree/master/test/example/example_djvu_xml [last accessed 9 June 2014].
11. **Hamilton, G** 2011 Post Office Directory Parser (podParser) PyPI podParser v0.3 package documentation. Available at http://pythonhosted.org/podparser/ [last accessed 20 February 2013].
12. **Hamilton, G** 2012 test.py. In Github: podparser/test. Available at https://github.com/edina/podparser/blob/master/test/tests.py [last accessed 9 June 2014].
13. **EDINA** 2013 AddressingHistory: Contact Us. Available at http://addressinghistory.edina.ac.uk/contactUs [last accessed 1 November 2013].
14. **JISC** 2010 Developing Community Content. Available at http://www.jisc.ac.uk/whatwedo/programmes/digitisation/communitycontent.aspx [last accessed 20 February 2013].

How to cite this article: Osborne, N, Hamilton, G and Macdonald, S 2014 Historical Post Office Directory Parser (POD Parser) Software From the AddressingHistory Project. *Journal of Open Research Software,* 2:e23, DOI: http://dx.doi.org/10.5334/jors.aq

Published: 21 July 2014

]u[*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS