The Journal of Open Source Software

Nominally: A Name Parser for Record Linkage

Matthew VanEseltine[1]

1 Institute for Social Research, University of Michigan

DOI: 10.21105/joss.03440

Software

- Review
- Repository
- Archive

Editor: Mark A. Jensen

Reviewers:

- @sara-02
- @sap218

Submitted: 19 May 2021
Published: 12 October 2021

Summary

With ever greater data availability, the importance of successfully connecting people across disparate datasets grows. As we link records from multiple sources, we would like to identify and measure similarities of names such as "Matthew VanEseltine" in one database, "Matt Van Eseltine" in another, and "Vaneseltine, M PhD" in a third. `Nominally` assists in initial stages of record linkage, where datasets are cleaned and preprocessed, by simplifying and parsing single-string personal names into six core fields: title, first, middle, last, suffix, and nickname.

Statement of Need

`Nominally` is a user-friendly Python package designed to parse large lists of names. It is independent of any specific data science framework and requires minimal dependencies. The `nominally` API provides simple command-line, function, and class access and easily integrates with the `pandas` (McKinney, 2010) data analysis library. The aim is to parse thousands or millions of strings into name parts for record linkage that maintain relevant features while excluding irrelevant details.

Human names can be difficult to work with in data. Varying quality and practices across institutions and datasets introduce noise and cause misrepresentation, increasing linkage and deduplication challenges. Common errors and discrepancies include (and this list is by no means exhaustive):

- Arbitrarily split first and middle names.
- Misplaced prefixes of last names such as "van" and "de la."
- Multiple last names partitioned into middle name fields.
- Titles and suffixes variously recorded in different fields, with or without separators.
- Inconsistent capture of accents, the ʻokina, and other non-ASCII characters.
- Single name fields arbitrarily concatenating name parts.

Cumulative variations and errors can combine to make the seemingly straightforward job of simply identifying first and last names rather difficult. `Nominally` is designed to consistently extract key features of personal names using a rule-based system (Christen, 2012). No prior differentiation is assumed between name fields; that is, `nominally` operates under the least informative case where only a single string name field is available. `Nominally` aggressively cleans input; scrapes titles, nicknames, and suffixes; and parses apart first, middle, and last names.

In its simplest application, `nominally` parses one name string into a dictionary of segmented name fields:

```
>>> from nominally import parse_name
>>> parse_name("Vimes, jr, Mr. Samuel 'Sam'")
{
    'title': 'mr',
    'first': 'samuel',
    'middle': '',
    'last': 'vimes',
    'suffix': 'jr',
    'nickname': 'sam'
}
```

Possible combinations of name parts are too extensive to itemize, but as a further example nominally extracts appropriate and comparable fields from these divergent presentations of a single name:

Input	Title	First	Middle	Last	Suffix	Nickname
S.T. VIMES JUNIOR		s	t	vimes	jr	
Vimes, Samuel T.		samuel	t	vimes		
samüél t vimés		samuel	t	vimes		
Samuel “sam” Thomas Vimes		samuel	thomas	vimes		sam
Dr. Samuel Thomas Vimes, Ph.D.	dr	samuel	thomas	vimes	phd	
Samuel T. Vimes, Jr. 24601		samuel	t	vimes	jr	
vimes, jr. phd, samuel		samuel		vimes	jr phd	

Nominally is designed for large-scale work. We employ nominally as part of record linkage in building the UMETRICS data at the Institute for Research on Innovation & Science (IRIS, 2020), which involves processing millions of name records of university employees, principal investigators, and published authors.

Comparisons with Existing Software

Multiple open-source Python packages focus on parsing names, including python-nameparser (Gulbranson, 2020), probablepeople (DataMade, 2019), and name-cleaver (Sunlight Labs, 2013). Nominally improves upon these packages in its core use case: parsing single human names of Western name order (first middle last). Nominally began from a fork of python-nameparser, initially aiming to refactor code and improve certain test cases. Development continued through a complete overhaul, and nominally now accurately handles a greater range of names without requiring user customization. Probablepeople and name-cleaver both cast a wider net, simultaneously addressing capture of multiple names, politicians, or companies. By narrowing the scope to single human names, nominally loses the broader applications of these packages but gains accuracy in its core capacity.

Large-scale data systems tend to impose a great many assumptions about the form and features of human names (McKenzie, 2010). As part of linking such systems together, nominally necessarily works within some such assumptions. Nominally does not attempt to identify a correct or ideal name, but rather to generate useful features of names using Western name order. Not all names can be accurately captured, and not all errors can be corrected, but many variations can be productively aligned.

Acknowledgements

Special thanks go to IRIS staff at the University of Michigan, who have run `nominally` at scale, provided feedback, and reported bugs. `Nominally` is indebted to the foundation of the `python-nameparser` project; its base of tests and name lists have been especially helpful throughout `nominally`'s development.

References

Christen, P. (2012). Data pre-processing. In *Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection* (pp. 39–67). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31164-2_3

DataMade. (2019). Probablepeople. In *GitHub repository*. https://github.com/datamade/probablepeople; GitHub.

Gulbranson, D. (2020). Python-nameparser. In *GitHub repository*. https://github.com/derek73/python-nameparser; GitHub.

IRIS. (2020). *IRIS UMETRICS 2020 linkage files*. https://doi.org/10.21987/70kd-x544

McKenzie, P. (2010). *Falsehoods programmers believe about names*. https://www.kalzumeus.com/2010/06/17/falsehoods-programmers-believe-about-names/

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Sunlight Labs. (2013). Name-cleaver. In *GitHub repository*. https://github.com/sunlightlabs/name-cleaver; GitHub.