The Journal of Open Source Software

PID.POS: An R package for the detection of personally identifiable data

Robert M. Cook [1], **Md Asaduzzaman** [1], **and Sarahjane Jones** [1]

1 University of Staffordshire, Centre for Health Innovation, Blackheath Lane, Stafford, England

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Adithi R Upadhya

Reviewers:

- @agerada
- @sebasquirarte

Submitted: 08 October 2025
Published: unpublished

Summary

The PID.POS package is designed to aid with the identification of personal identifiability risks in data sets. By applying existing natural language processing techniques, the package is able to identify proper nouns within a data set. The extraction of proper nouns reduced the complexity of the data, allowing for a quicker review and oversight of the data. The package also includes a basic tool for the design, and implementation of a redaction process.

Statement of need

The world is embedded in a data revolution. Never before have we had the depth or breadth of data being captured and analysed than we do at present, and this is only set to increase. In response, international bodies are taking steps to ensure legal protection of an individual's rights to their own data (European Parliament & Council of the European Union, 2016). One effect of increase legislation in the European Union has been a growing awareness of the role and responsibility of data controllers (ICO, n.d.-b) and the risks of big data (Clarke, 2016). Among these concerns, a risk of 'personal identifiability' i.e. the ability to directly or indirectly identify an individual from a dataset (Finck & Pallas, 2020), is paramount and, if breeched, can lead to reputation damage and fines (ICO, n.d.-a).

Where data is structured and comprises only a few hundred observations, a manual inspection can identify variables which contain directly personally identifiable data (PID) with a reasonable time investment. However, in the case of modern large data sets which may comprise millions of observations, a manual inspection may miss PID if it is embedded within a passage of text, or is a rarity for the given variable. The PID.POS (Personal Identifiability Detection by Part Of Speech tagging) package is designed to aid with the identification of PID risks in data sets. In comparison to existing packages which rely on a curated list of common names and string-matching, PID.POS builds on the existing udpipe framework(Straka et al., 2016; Wijffels, 2023), extracting all examples of proper nouns and providing a mechanism for the review and redaction of PID risks.

Comparison to existing R packages

The need to review data sets to identify risks is not new, and there are a number of packages which have been developed to aid in this process. The most notable of these are the PII package (Patterson-Stein, 2025), which is designed to identify personally identifiable features via pattern matching. These approaches can be effective in identifying PID, but have a risk of missing edge cases e.g. relying on sentence case to identify names. The approach taken in PID.POS conversely takes the approach of purposefully extracting all proper-nouns, and hence increase the false positive rate, with the intention of supplying a simplified extract to aid human interpretation rather than fully automate it.

The Journal of Open Source Software

In practice

To install the current version of `PID.POS` package, use the following code:

```
# install.packages("pak")
pak::pkg_install("Stat-Cook/PID.POS")
```

To assist with understanding the `PID.POS` package, we include a subset of the 'friends' data set from the `friends` package.

```
library(pid.pos)
the_one_in_massapequa
```

scene	utter-ance	speaker	text
1	1	Scene Directions	[Scene: Central Perk, everyone is there.]
1	2	Phoebe Buffay	Oh, Ross, Mon, is it okay if I bring someone to your parent's anniversary party?
1	3	Monica Geller	Yeah.
1	4	Ross Geller	Sure. Yeah.
1	5	Joey Tribbiani	So, who's the guy?

The package has two main functions for identifying PID risks, depending on the users needs. First, the `data_frame_report` function converts a typical R data frame into a new data frame of:

- `ID` - the column and row where the sentence first appears
- `Token` - the specific proper noun token
- `Sentence` - the sentence containing proper nouns
- `Repeats` - the number of times the sentence occurs in the data set
- `Affected Columns` - the columns in the original data frame which contain the sentence

```
report <- data_frame_report(the_one_in_massapequa)
report
```

ID	Token	Sentence	Document	Repeats	Affected Columns
Col:speaker Row:2	Phoebe	Phoebe Buffay	Phoebe Buffay	40	speaker
Col:speaker Row:2	Buffay	Phoebe Buffay	Phoebe Buffay	40	speaker
Col:speaker Row:3	Monica	Monica Geller	Monica Geller	25	speaker
Col:speaker Row:3	Geller	Monica Geller	Monica Geller	25	speaker
Col:speaker Row:4	Ross	Ross Geller	Ross Geller	43	speaker

For a top level summary of the report, the `summary` method for class `pid_report` can be used:

```
summary(report)
```

Column	Cases of Proper Nouns	Unique Cases of Proper Nouns	Most Common Proper Noun Sentence
speaker	243	14	Ross Geller
text	99	99	[Scene: Central Perk, everyone is there.]

The second function is `report_on_folder` which iterates over a folder of data files, producing a proper noun report for each. It is foreseen that this function will be the more useful, used just before data release to evidence no PID risks.

```
report_on_folder('path/to/data/')
browse_model_location()
```

NB: the `data_frame_report` and `report_on_folder` functions automate the download of the pre-trained `udpipe` model. These models are required to be cached to the users hard-drive and hence firewall issues may present. The vignette … is included to help with common issues.

While being able to identify PID risks is the core premise of this package, it would be remiss to not supply some tools to aid in the removal of PID. Hence, we supply
basic functionality designed for minimal technical knowledge to assist in the redaction of PID.

Where a PID report has been ran, the resulting data frame can be passed to the function `report_to_redaction_rules` which will convert the report to a csv file with three headings:

- `If` - the sentence pattern which, if it matches, the replacement is applied
- `From` - the pattern to be replaced
- `To` - the intended replacement

```
replacement_rules <- report_to_redaction_rules(
  report,
  path='path/to/report.csv'
)
```

If	From	To
Phoebe Buffay	Phoebe	Phoebe
Phoebe Buffay	Buffay	Buffay
Monica Geller	Monica	Monica
Monica Geller	Geller	Geller
Ross Geller	Ross	Ross

The csv file is intended to be edited by the data controller, who hence does not need to understand R, and can be reimported using the `prepare_redactions` function:

```
prepare_redactions('path/to/report.csv')
```

The `prepare_redactions` function creates a string replacement rule to capture the desired redactions, with the option for R to 'parse' the function for use as part of a data pipeline:

```
redaction.func <- prepare_redactions('path/to/report.csv')

the_one_in_massapequa |>
  mutate(
    across(
      where(is.character),
      redaction.func
    )
  )
```

Further utilities are available, notably tools to automatically encode the `To` column (see Auto Replacements).

Current applications

The `PID.POS` package was developed for applications in the NuRS and AmReS research projects which aim to extract and analyse retrospective operational data from NHS Trusts to understand staff retention and patient safety.

Contributions

The package was designed by RC, MA and SJ. Implementation was done by RC. Quality assurance was done by MA. Documentation was written by RC. Funding for the work was won by RC and SJ.

Acknowledgements

The development of `PID.POS` was part of the NuRS and AmReS projects funded by the Health Foundation.

References

Clarke, R. (2016). Big data, big risks. *Information Systems Journal*, *26*(1), 77–90.

European Parliament, & Council of the European Union. (2016, April 27). *Regulation (EU) 2016/679 of the european parliament and of the council*. https://data.europa.eu/eli/reg/2016/679/oj

Finck, M., & Pallas, F. (2020). They who must not be identified—distinguishing personal from non-personal data under the GDPR. *International Data Privacy Law*, *10*(1), 11–36.

ICO. (n.d.-a). *Personal data breaches: What happens if we fail to notify the ICO of all notifiable breaches?* https://ico.org.uk/for-organisations/report-a-breach/personal-data-breach/personal-data-breaches-a-guide/#whathappensi

ICO. (n.d.-b). *What does it mean if you are a controller?* https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/controllers-and-processors/controllers-and-processors/what-does-it-mean-if-you-are-a-controller/

Patterson-Stein, J. (2025). *Pii: Search data frames for personally identifiable information*. https://CRAN.R-project.org/package=pii

Straka, M., Hajic, J., & Straková, J. (2016). UDPipe: Trainable pipeline for processing CoNLL-u files performing tokenization, morphological analysis, pos tagging and parsing. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 4290–4297.

Wijffels, J. (2023). *Udpipe: Tokenization, parts of speech tagging, lemmatization and dependency parsing with the 'UDPipe' 'NLP' toolkit*. https://CRAN.R-project.org/package=udpipe