

textNet: Directed, Multiplex, Multimodal Event Network Extraction from Textual Data

Elise Zufall¹ and Tyler Scott¹

¹ UC Davis Department of Environmental Science and Policy

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Open Journals](#) ↗

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright
and release the work under a
Creative Commons Attribution 4.0
International License ([CC BY 4.0](#)).

Introduction

Network measurement in social science typically relies on data collected through surveys and interviews. Document-based measurement is automatable and scalable, providing opportunities for large scale or longitudinal research that are not possible through traditional methods. A number of tools exist to generate networks based on co-occurrence of words within documents (such as the [Nocodefunctions](#) app ([Levallois et al., 2012](#)), the “[textnets](#)” package ([Bail, 2024](#)), [InfraNodus](#) ([Paranyushkin, 2018](#)), and many more). But there is, to our knowledge, no open-source tool that generates network data based on the syntactic relationships between entities within a sentence. *textNet* allows a user to input one or more PDF documents and create arbitrarily complex directed, multiplex, and multimodal network graphs. *textNet* also works on arbitrarily long documents, making it well suited for research applications using long texts such as government planning documents, court proceedings, regulatory impact analyses, and environmental impact assessments.

Statement of Need

Network extraction from documents has typically required manual coding. Furthermore, existing network extraction methods that use co-occurrence leave a vast amount of data on the table, namely, the rich edge attribute data and directionality of each verb phrase defining the particular relationship between two entities, and the respective roles of the entity nodes involved in that verb phrase. We present an R package, *textNet*, designed to enable directed, multiplex, multimodal network extraction from text documents through syntactic dependency parsing, in a replicable, automated fashion for collections of arbitrarily long documents. The *textNet* package facilitates the automated analysis and comparison of many documents, based on their respective network characteristics. Its flexibility allows for any desired entity categories, such as organizations, geopolitical entities, dates, or custom-defined categories, to be preserved.

Directed Graph Production

As a syntax-based network extractor, *textNet* identifies source and target nodes. This produces directed graphs that contain information about network flow. Methods based on identifying co-occurring nodes in a document, by contrast, produce undirected graphs. Co-occurrence graphs also have the tendency to generate saturated subgraphs, since every co-occurring collection of entities has every possible edge drawn amongst them. By contrast, *textNet* draws connections not between every entity in the document or even the sentence, but specifically between pairs of entities that are mediated by an event relationship.

Multiplex Graph Output

Syntax-based measurement encodes edges based on subject-verb-object relationships. *textNet* stores verb information as edge attributes, which allows the user to preserve arbitrarily complex

40 topological layers (of different types of relationships) or customize groupings of edge types to
41 simplify representation.

42 Multimodal Graph Output

43 Multimodal networks, or networks where there are multiple categories of nodes, have common
44 use cases such as social-ecological network analysis of configurations of actors and environmental
45 features. Existing packages such as the *manynet* package (Hollway, 2024) provide analytical
46 functions for multimodal network statistics. *textNet* provides a structure for tagging and
47 organizing arbitrarily complex node labeling schemes that can then be fed into packages for
48 multi-node network statistical analysis. Node labels can be automated (e.g., the default entity
49 type tags for an NLP engine such as *spaCy* (Honnibal et al., 2021)), customized using a
50 dictionary, or based on a hybrid scheme of default and custom labels. Any node type is possible
51 (e.g., species, places, people, concepts, etc.) so this can be adapted to domain-specific research
52 applications by applying dictionaries or using a custom NER model.

53 Installation

54 The stable version of this package can be installed from Github, using the *devtools* package
55 (Wickham et al., 2022):

```
56 devtools::install_github("ucd-cep/textnet")
```

57 The *textNet* package suggests several convenience wrappers of packages such as *spacyr* (Benoit
58 et al., 2023), *pdftools* (Ooms, 2024), *igraph* (Csárdi et al., 2024), and *network* (Butts et al.,
59 2023). To use the full functionality of *textNet*, such as pre-processing tools and post-processing
60 analysis tools, we recommend installing these packages, which for *spacyr* requires integration
61 with Python. However, the user may wish to preprocess and parse data using their own NLP
62 engine, and skip directly to the `textnet_extract()` function, which does not depend on *spacyr*
63 or Python integration.

64 Overview and Main Functions

65 The package architecture relies on four sets of functions around core tasks:

- 66 ■ [OPTIONAL] Pre-processing: `pdf_clean()`, a wrapper for the `pdftools::pdf_text()` func-
67 tion which includes a custom header/footer text removal feature; and `parse_text()`,
68 which is a wrapper for the *spacyr* package and uses the *spaCy* natural language pro-
69 cessing engine (Honnibal et al., 2021) to parse text and perform part of speech tagging,
70 dependency parsing, and named entity recognition (NER). Alternatively, the user can
71 skip this step and load parsed text directly into the package. Externally produced data
72 must be converted to standards outlined in the package manual.
- 73 ■ Network extraction: `textnet_extract()`, which generates a graph database from parsed text
74 based upon tags and dependency relations. The object returned from `textnet_extract()`
75 consists of a *nodelist*, an *edgelist* with a rich set of edge attributes, a *verblist*, and a list
76 of potential coreferences for disambiguation.
- 77 ■ Disambiguation: tools for cleaning, recoding, and aggregating node and edge attributes,
78 such as the `find_acronyms()` function, which can be paired with the `disambiguation()`
79 function to identify acronyms in the text and replace them with the full entity name.
- 80 ■ Exploration: the `export_to_network()` function for exporting the graph database to
81 *igraph* and *network* objects, `top_features()` for viewing node and edge attributes, and
82 `combine_networks()` for aggregating multiple document-based graphs based on common
83 nodes.

84 The figure below summarizes the functionality of *textNet* and the flow of function outputs.
85 Optional data cleaning features are shown with dotted arrows.



Figure 1: Workflow of textNet Functions

Applications

textNet has applications in governance network scholarship, as demonstrated by Zufall and Scott (2024) and by ongoing work on groundwater governance at the UC Davis Center for Environmental Policy and Behavior. Additional potential applications include legal scholarship, social-ecological network analysis, government planning documents, court proceedings, archival research, communication and media research, and other fields interested in exploring events and entity relationships in textual data.

Potential Further Analyses

textNet is compatible with standard network analysis tools in R. Functionality from *ggraph* (Pedersen & RStudio, 2024), *sna* (Butts, 2024), *igraph* (Csárdi et al., 2024), *network* (Butts et al., 2023), and other network visualization and analysis packages can be used to further explore the extracted networks.

The *ggraph* package has been used to create the network visualization seen here, using a weighted version of a *igraph* constructed using the “old_new_parsed” sample data in *textNet*.

New Network



Figure 2: Representation of the Event Network of the New Plan

100 The network-level attributes output from `export_to_network` can also be analyzed against
101 exogenous metadata that has been collected separately by the researcher regarding the different
102 documents and their real-world context. The extracted networks can also be analyzed through
103 a variety of tools, such as an Exponential Random Graph Model or a Temporal Exponential
104 Random Graph Model.

105 Vignette

106 More information about the entity network extraction algorithm and an example start-to-finish
107 data processing and analysis workflow can be found in the vignette for this package. The
108 vignette uses sample data that travels with the *textNet* package.

109 Acknowledgements

110 The authors gratefully acknowledge the support of the Sustainable Agricultural Systems
111 program, project award no. 2021-68012-35914, from the U.S. Department of Agriculture's
112 National Institute of Food and Agriculture and the National Science Foundation's Dynamics
113 of Integrated Socio-Environmental Systems program, grant no. 2205239.

114 References

- 115 Bail, C. (2024). *Cbail/textnets* (Version 0.1.1). <https://github.com/cbail/textnets>
- 116 Benoit, K., Matsuo, A., Gruber, J., & Council (ERC-2011-StG 283794-QUANTESS), E. R.
117 (2023). *Spacyr: Wrapper to the 'spaCy' 'NLP' library* (Version 1.3.0). <https://cran.r-project.org/web/packages/spacyr/index.html>
- 118 Butts, C. T. (2024). *Sna: Tools for social network analysis* (Version 2.8). <https://cran.r-project.org/web/packages/sna/index.html>
- 119 Butts, C. T., Hunter, D., Handcock, M., Bender-deMoll, S., Horner, J., Wang, L., Krivitsky, P.
120 N., Knapp, B., Bojanowski, M., & Klumb, C. (2023). *Network: Classes for relational data*
121 (Version 1.18.2). <https://cran.r-project.org/web/packages/network/index.html>
- 122 Csárdi, G., Nepusz, T., Traag, V., Horvát, S., Zanini, F., Noom, D., Müller, K., Salmon, M.,
123 Antonov, M., & details, C. Z. I. igraph author. (2024). *Igraph: Network analysis and*
124 *visualization* (Version 2.1.1). <https://cran.r-project.org/web/packages/igraph/index.html>
- 125 Hollway, J. (2024). *Manynet: Many ways to make, modify, map, mark, and measure myriad*
126 *networks* (Version 1.2.6). <https://CRAN.R-project.org/package=manynet>
- 127 Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2021). *spaCy: Industrial-strength*
128 *natural language processing in python* (Version 3.1.3). <https://github.com/explosion/spaCy/tree/master>
- 129 Levallois, C., Clithero, J. A., Wouters, P., Smidts, A., & Huettel, S. A. (2012). Translating
130 upwards: Linking the neural and social sciences via neuroeconomics. *Nature Reviews*
131 *Neuroscience*, 13(11), 789–797. [https://nocodefunctions.com/cowo/semantic_networks_](https://nocodefunctions.com/cowo/semantic_networks_tool.html)
132 [tool.html](https://nocodefunctions.com/cowo/semantic_networks_tool.html)
- 133 Ooms, J. (2024). *Pdftools: Text extraction, rendering and converting of PDF documents*
134 (Version 3.4.1). <https://cran.r-project.org/web/packages/pdftools/index.html>
- 135 Paranyushkin, D. (2018). *InfraNodus*. Nodus Labs. <https://infranodus.com/>
- 136 Pedersen, T. L., & RStudio. (2024). *Ggraph: An implementation of grammar of graphics for*
137 *graphs and networks* (Version 2.2.1). [https://cran.r-project.org/web/packages/ggraph/](https://cran.r-project.org/web/packages/ggraph/index.html)
138 [index.html](https://cran.r-project.org/web/packages/ggraph/index.html)
- 139 Wickham, H., Hester, J., Chang, W., Bryan, J., & RStudio. (2022). *Devtools: Tools to make*
140 *developing r packages easier* (Version 2.4.5). <https://cran.r-project.org/web/packages/>

144 [devtools/index.html](#)

145 Zufall, E., & Scott, T. A. (2024). Syntactic measurement of governance networks from
146 textual data, with application to water management plans. *Policy Studies Journal*, n/a.
147 <https://doi.org/10.1111/psj.12556>

DRAFT