


# text2map: R Tools for Text Matrices

Dustin S. Stoltz<sup>\*1</sup> and Marshall A. Taylor<sup>†2</sup>

1 Lehigh University 2 New Mexico State University

DOI: [10.21105/joss.03741](https://doi.org/10.21105/joss.03741)

## Software

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

---

Editor: [Chris Hartgerink](#) 

## Reviewers:

- [@alexanderfurnas](#)
- [@cmaimone](#)

Submitted: 25 August 2021

Published: 20 April 2022

## License

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

## Summary

text2map is an R ([R Core Team, 2021](#)) package that provides several tools for working with text matrices, including document-term matrices, term-context matrices, and word embedding matrices. text2map is published at The Comprehensive R Archive Network (CRAN) at <https://cran.r-project.org/package=text2map>; its source is available at <https://gitlab.com/culturalcartography/text2map/>. text2map contains vignettes demonstrating basic functionality and more advanced uses of the package. The website, which includes function documentation, is available at <https://culturalcartography.gitlab.io/text2map/>.

Specifically, text2map offers functions for creating semantic centroids, semantic regions, semantic directions, semantic projections, transforming embeddings, and performing concept mover’s distance and concept class analysis ([Arseniev-Koehler et al., 2021](#); [Arseniev-Koehler & Foster, 2020](#); [Boutyline et al., 2020](#); [Caliskan et al., 2017](#); [Carbone & Mijs, 2022](#); [Jones et al., 2020](#); [Nelson, 2021](#); [Stoltz & Taylor, 2019, 2021](#); [Taylor & Stoltz, 2020a, 2020b](#)).

Embeddings often need to be transformed after training. `find_transformation` provides methods for normalizing and centering large embedding matrices. Embedding matrices which have been trained separately can also be “aligned” using `find_transformation` using the Procrustes method. Embeddings can also be “projected” onto a given vector or “rejected” from a given vector using `find_projection` and `find_rejection` respectively.

**Semantic centroids** are increasingly used to “fine-tune” pretrained embedding vectors. text2map takes a vector of “anchor” terms and finds the vector average for those terms. This allows the user to specify a given sense of a word or concept. Similarly, **semantic directions** are used to define a one-dimensional subspace corresponding to a bipolar concept – such as “big” to “small” ([Grand et al., 2022](#)) or “conservative” to “liberal” ([Taylor & Stoltz, 2020b](#)). The most well known example is “gender,” where masculine terms anchor one side of the direction and feminine terms anchor the other. text2map offers four methods for defining a semantic direction: paired (each individual term is subtracted from exactly one other paired term), pooled (terms corresponding to one side of a direction are first averaged, and then these averaged vectors are subtracted), L2 (the direction is calculated the same as with “pooled” but is then divided by the L2 Euclidean norm), and PCA (vector offsets are calculated for each pair of terms, as with “paired,” and if `n_dirs = 1L` (the default) then the direction is the first principal component). With the latter option, users can return more than one direction. text2map also provides a range of precompiled anchor lists from published works, defining 26 semantic relations, including gender and status. We should emphasize that these should be used as a starting point, and not as “ground truth.”

In addition, text2map provides tools related to the document-term matrix (DTM), including a fast unigram DTM builder and comprehensive DTM “stopping” function, both of which are built around the Matrix class `dgCmatrix`. The package’s DTM resampler function takes any

---

\*co-first author

†co-first author

DTM and randomly resamples from each row, creating a new DTM randomly. This can be useful in many downstream tasks.

## Statement of Need

`text2map` offers a consistent set of tools built around representing texts as matrices. This is in contrast to corpus objects (Perry, 2021) or `tidytext`'s triplet data frame (Silge & Robinson, 2016). This allows `text2map` to remain close to the underlying matrix mathematics of contemporary computational text analysis as well as make use of memory-efficient matrix packages—e.g., `Matrix` (Bates & Maechler, 2010). `text2map` also avoids special-purpose classes, such as `quanteda`'s `dfm` class. While there are R packages for training word embeddings—e.g., `text2vec` (Selivanov et al., 2020)—none offer methods for working with embeddings in downstream tasks, in particular, tasks involved in social scientific and digital humanities research. For example, to the best of our knowledge, `text2map` is the only R package that provides functions dedicated to finding semantic centroids, semantic directions, and embedding matrix transformations. As such, packages such as `text2vec` can be used to train embeddings, and `text2map` can then be used to ready the embeddings for downstream analyses. Further, while `quanteda` is a general purpose suite of tools built around the corpus object, `text2map` focuses on the matrix as the central data object.

## Illustration

Let's consider a simple example of `text2map` in use in a humanities context. Building off some of our previous work (Stoltz & Taylor, 2019; Taylor & Stoltz, 2020b), say a researcher is interested in examining the extent to which Shakespeare's First Folio plays engage the concept of "death."

The `text2map` package can be used to efficiently convert the raw corpus of plays into a document-term matrix (DTM), compute several different types of summary statistics on that DTM, and then measure each plays engagement with the concept of interest. The package will also work with DTM built from other popular text analysis packages.

Using `dtm_stats` we can get a series of summary statistics. The table below illustrates potential output (`hapax`, `dis`, `tris` refers to terms occurring just once, twice, and thrice, respectively, in the corpus).

**Table 1:** Basic DTM Information

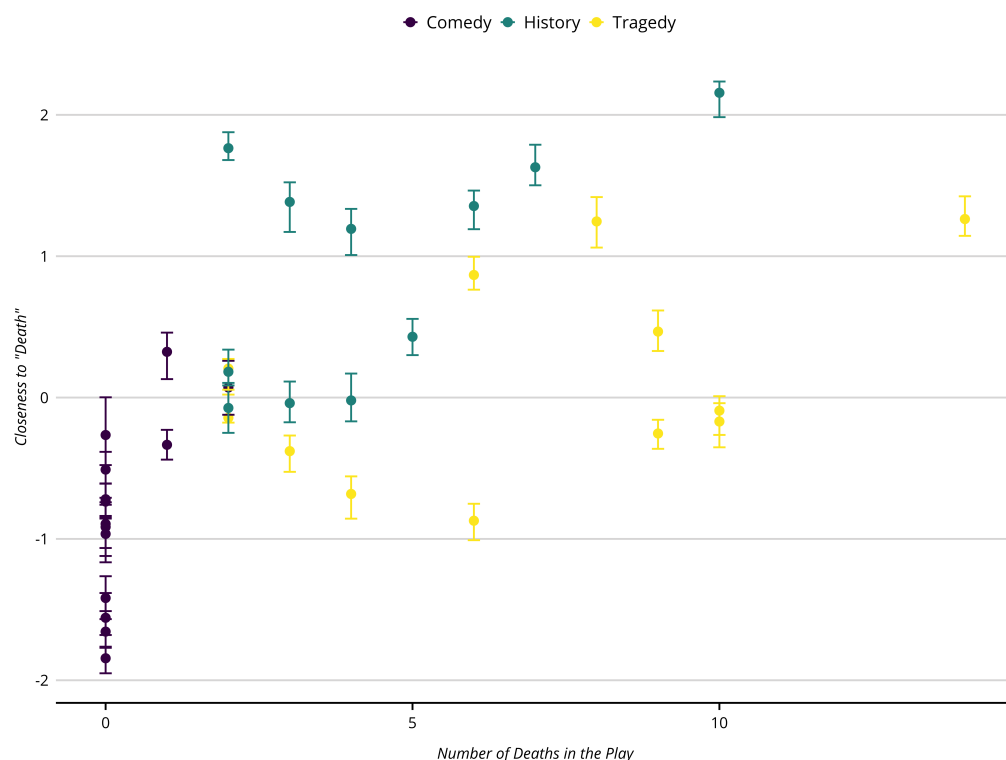
	Measure	Value
1	Total Docs	37
2	Percent Sparse	87.70%
3	Total Types	29957
4	Total Tokens	889428
5	Object Size	3.6 Mb

**Table 2:** Lexical Richness Metrics

	Measure	Value
1	Percent Hapax	47.00%
2	Percent Dis	25.00%
3	Percent Tris	21.00%
4	Type-Token Ratio	0.03

We can then quickly generate concept mover's distance (CMD) scores using the `CMDist` function and a matrix of word vectors. The user can also add "sensitivity intervals" shown in the plot using the package's `dtm_resampler` function to assess how robust each production's CMD score is (or is not) to the specific vocabulary frequency distribution of that document.

In the example, the CMD of each play to the single word "death" is computed, however, the user could use `get_centroid` to specify death using a few synonyms. We could also compute each play's CMD to death *as opposed to* life by using the output of `get_direction` and pairs of anchors for life and death, respectively.



**Figure 1:** Illustrative figure. Scatterplot of Shakespeare play's CMD scores (*y*-axis) and the body count in the narrative (*x*-axis). Bands are sensitivity intervals, which are the CMD scores at the 2.5 and 97.5 percentiles for each document after resampling the vocabulary from the document 20 times.

## Acknowledgements

We would like to thank Michael Lee Wood for helpful advice and test-runs with the software. We would also like to thank Brandon Sepulvado for his assistance in fixing a parallelization error in earlier iterations of the `CMDist` code.

## References

- Arseniev-Koehler, A., Cochran, S. D., Mays, V. M., Chang, K.-W., & Foster, J. G. (2021). *Integrating topic modeling and word embedding to characterize violent deaths*. <https://doi.org/10.31235/osf.io/nkyaq>
- Arseniev-Koehler, A., & Foster, J. G. (2020). *Machine learning as a model for cultural learning: Teaching an algorithm what it means to be fat*. <https://doi.org/10.31235/osf.io/c9yj3>
- Bates, D., & Maechler, M. (2010). *Matrix: Sparse and dense matrix classes and methods*.

- Boutyline, A., Arseniev-Koehler, A., & Cornell, D. (2020). School, studying, and smarts: Gender stereotypes and education across 80 years of american print media, 1930-2009. In *SocArxiv*. <https://doi.org/10.31235/osf.io/bukdg>
- Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), 183–186. <https://doi.org/10.1126/science.aal4230>
- Carbone, L., & Mijs, J. (2022). Sounds like meritocracy to my ears: Exploring the link between inequality in popular music and personal culture. *Information, Communication and Society*, 1–19. <https://doi.org/10.1080/1369118X.2021.2020870>
- Grand, G., Blank, I. A., Pereira, F., & Fedorenko, E. (2022). Semantic projection recovers rich human knowledge of multiple object features from word embeddings. *Nature Human Behaviour*, 1–13. <https://doi.org/10.1038/s41562-022-01316-8>
- Jones, J. J., Amin, M. R., Kim, J., & Skiena, S. (2020). Stereotypical gender associations in language have decreased over time. *Sociological Science*, 7(1), 1–35. <https://doi.org/10.15195/v7.a1>
- Nelson, L. K. (2021). Leveraging the alignment between machine learning and intersectionality: Using word embeddings to measure intersectional experiences of the nineteenth century U.S. south. *Poetics*, 88, 101539. <https://doi.org/10.1016/j.poetic.2021.101539>
- Perry, P. O. (2021). *Corpus: Text corpus analysis*. <https://CRAN.R-project.org/package=corpus>
- R Core Team. (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Selivanov, D., Bickel, M., & Wang, Q. (2020). *text2vec: Modern text mining framework for R*.
- Silge, J., & Robinson, D. (2016). Tidytext: Text mining and analysis using tidy data principles in r. *JOSS*, 1(3). <https://doi.org/10.21105/joss.00037>
- Stoltz, D. S., & Taylor, M. A. (2019). Concept mover's distance. *Journal of Computational Social Science*, 2, 293–313. <https://doi.org/10.1007/s42001-019-00048-6>
- Stoltz, D. S., & Taylor, M. A. (2021). Cultural cartography with word embeddings. *Poetics*, 101567. <https://doi.org/10.1016/j.poetic.2021.101567>
- Taylor, M. A., & Stoltz, D. S. (2020a). Concept class analysis: A method for identifying cultural schemas in texts. *Sociological Science*, 7(23), 544–569. <https://doi.org/10.15195/v7.a23>
- Taylor, M. A., & Stoltz, D. S. (2020b). Integrating semantic directions with concept mover's distance to measure binary concept engagement. *Journal of Computational Social Science*, 4, 231–242. <https://doi.org/10.1007/s42001-020-00075-8>