The Journal of Open Source Software

anvay: A Web-based Tool for Interpretive Topic Modelling in Bengali

Vinayak Das Gupta [1]

1 Shiv Nadar Institution of Eminence

DOI: 10.21105/joss.08641

Software

- Review
- Repository
- Archive

Editor: Abhishek Tiwari

Reviewers:

- @dharanpreethi
- @SouravGhosh-digital-humanities
- @x-tabdeveloping

Submitted: 22 April 2025
Published: 17 February 2026

Summary

anvay is a web-based tool for topic modelling in Bengali, developed for exploratory reading and interpretive analysis. It provides a full pipeline for Latent Dirichlet Allocation (LDA) (Blei et al., 2003)—from corpus ingestion and preprocessing to model configuration and visual output—within a lightweight, browser-based interface. The tool foregrounds user interpretation: rather than providing coherence scores or fixed topic labels, *anvay* presents the model output to be read, interpreted, and adjusted by the user.

Designed for literary, journalistic, and historical corpora in Bengali, *anvay* supports a range of language-specific preprocessing functions including lemmatisation, frequency filtering, and stopword pruning. The outputs, ranging from topic-word networks to document-level previews, are rendered with clarity and designed to enable close reading. Each topic is accessible through multiple lenses: top words, paragraph-level examples, document weights, and corpus-wide distribution.

Beyond its technical function, *anvay* is an intervention in how we teach and understand computational methods within the humanities in low-resource contexts.

State of the Field

Topic modelling is well established in the digital humanities and is used for analysing large text collections in literary studies, cultural analytics, and media research (Goldstone & Underwood, 2014b; Jockers, 2013). Scholars employ models to identify recurring themes and support exploratory reading.

Gensim (Řehůřek & Sojka, 2010) and Mallet (McCallum, 2002) are popular backend libraries. Several interfaces support this work, and *anvay* draws inspiration from many of these. Voyant Tools provides a general-purpose environment for text analysis, which includes an accessible topic-modelling component (Sinclair & Rockwell, 2016). Termite, one of the earliest contributions of this type, offers a clear tabular display for comparing topic–term relations (Chuang et al., 2012). pyLDAvis supplies interactive views of topic distances and word relevance (Sievert & Shirley, 2014). TopicWizard presents topic clusters, keywords and document-level patterns through a web interface (Kardos et al., 2025). jsLDA demonstrates that browser-based topic modelling is feasible and useful in teaching contexts (Mimno, n.d.). Other platforms have made model outputs available to wider audiences. The Topic Modeling Tool provides a simple graphical front end to MALLET (Enderle, 2019). DFR-Browser supports exploratory reading of topic models within journal archives (Goldstone & Underwood, 2014a).

Recent systems such as BERTopic combine contextualised representations with neural topic models to improve coherence (Grootendorst, 2022). Other methods, such as Top2Vec (Angelov, 2020) and CombinedTM (Bianchi et al., 2021) built on sentence transformers or contextualised embeddings, also aim to improve topic coherence and reduce dependence on bag-of-words features.

Statement of Need

Most topic-modelling tools are designed for English and other high-resource languages. They rely on tokenisers, stemmers and embedding models that do not transfer well to Bengali. They might also require users to prepare their own pipelines or rely on English-centric defaults.

Researchers and students working with Bengali texts face several difficulties. Tokenisation may produce malformed units, existing stopword lists are incomplete and lemmatisation resources are limited. In teaching settings, students often lack the technical background to run scripts or to interpret model output without guidance.

Transformer-based topic-modelling systems, such as BERTopic, Top2Vec and contextual topic models, are not suitable for this project. They rely on pretrained embeddings that do not exist for many Bengali registers. They are slower and less lightweight than classical LDA, which limits accessibility in browser-based or workshop environments. They also behave as opaque models, which conflicts with the interpretive aims of the interface.

anvay addresses these problems by offering a full topic-modelling workflow tailored to Bengali. It provides appropriate tokenisation and lemmatisation, configurable model parameters and a set of visualisations that foreground interpretability. Users can explore top words, representative sentences and topic distributions directly in the browser. The tool also supplies documentation that helps students understand how topic models operate and how to read them critically.

Functionality

anvay is written in Python and uses Flask, Gensim, and standard visualisation libraries. Its main features include:

- **Corpus upload and cleaning**: Up to 800 `.txt` files can be uploaded. Users can apply stopword filters with NLTK (Bird et al., 2009) or user stopword lists, stemming or lemmatisation, and token pruning.
- **Model training**: Parameters such as passes, iterations, alpha, and chunk size can be adjusted. Models are trained on the server.
- **Bengali processing**: Tokenisation avoids malformed output. Lemma data is drawn from public resources (Alam et al., 2021; Chakrabarty et al., 2017).
- **Visualisations**: Results are shown using:
  - Topic scatter plots (Plotly)
  - Heatmaps (Seaborn) (Waskom, 2021)
  - Bar and pie charts for topic-document relations
  - Topic-word network graphs (NetworkX) (Hagberg et al., 2008)
- **Interpretive tools**: Users can see representative paragraphs, find key topics in each file, and compare topic strength. Topics that appear noisy or incoherent are flagged with a “Low Confidence” warning.
- **Report generation**: Alongside visual outputs, *anvay* creates a structured report that prints the training configuration, dataset statistics, and top keywords per topic. This includes metrics like document and token counts, vocabulary size, topic prevalence, and topic weights per document. A representative sentence is also shown for each topic. These help users trace how the model was built and better understand its results.
- **Export and accessibility**: The tool supports CSV and TXT downloads. It works in all major browsers, with responsive design and dark mode.

Research and Pedagogical Use

The design of *anvay* is informed by research-led teaching practice. Topic modelling, while widely adopted in digital humanities, often remains inaccessible due to steep learning curves and underdeveloped interfaces. *anvay* was developed to lower these barriers and support new modes of engagement with Bengali textual corpora, especially where existing NLP tools fail to account for morphological variance, informal orthographies, or the diversity of textual registers in Bengali.

In pedagogical contexts, *anvay* functions as a conceptual primer. It prompts students to ask: What is a topic? What assumptions shape a model's output? How do visualisations shape interpretation? The tool has been tested in classroom environments with undergraduate and postgraduate students, many of whom were engaging with topic modelling for the first time. The feedback has been consistent: the visual design, language support, and document-level previews help to render the model's assumptions legible.

To support this, *anvay* includes extensive web-based documentation. Each section guides users through corpus preparation, parameter tuning, and result analysis, with annotated examples and embedded visual references. The documentation foregrounds conceptual understanding: users are encouraged to read models critically, experiment with settings, and reflect on how computational structure intersects with thematic interpretation. It is embedded directly in the interface and designed for both classroom and independent study.

Performance and Limitations

anvay is designed for moderate-scale corpora, where interpretability and visual exploration are prioritised over throughput. In a benchmark run using **800 Bengali `.txt` files** (totalling **21.9MB**, ~**940,000 tokens**, and **171,754 unique vocabulary terms**), the system successfully trained a 10-topic LDA model with **10 passes** and **50 iterations** in approximately **62 seconds** on a single-core setup. This corpus included highly variable document lengths, from **79** to **86,099 tokens** per file, demonstrating robustness against input heterogeneity.

While the system is tuned for formal Bengali prose, there are limitations: informal or dialectal orthographies may lead to malformed tokens, and OCR artefacts or non-Unicode glyphs may interfere with tokenisation.

The modelling backend is standard LDA; no coherence optimisation or neural alignment is included. As such, *anvay* is best used as an exploratory interface, for interpretive reading rather than automated evaluation.

Repository and License

The source code and documentation for *anvay* are hosted on GitHub: https://github.com/vinayakdasgupta
The software is released under the MIT License.

References

Alam, F., Hasan, Md. A., Alam, T., Khan, A., Tajrin, J., Khan, N., & Chowdhury, S. A. (2021). A review of Bangla natural language processing tasks and the utility of transformer models. *arXiv Preprint arXiv:2107.03844*. https://arxiv.org/abs/2107.03844

Angelov, D. (2020). *Top2Vec: Distributed representations of topics*. https://arxiv.org/abs/2008.09470

Bianchi, F., Terragni, S., & Hovy, D. (2021). Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In C. Zong, F. Xia, W. Li, & R. Navigli

(Eds.), *Proceedings of the 59th annual meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (volume 2: Short papers)* (pp. 759–766). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-short.96

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media, Inc.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Chakrabarty, A., Pandit, O. A., & Garain, U. (2017). Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In R. Barzilay & M.-Y. Kan (Eds.), *Proceedings of the 55th annual meeting of the Association for Computational Linguistics (volume 1: Long papers)* (pp. 1481–1491). Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1136

Chuang, J., Manning, C. D., & Heer, J. (2012). Termite: Visualization techniques for assessing textual topic models. *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 74–77. https://doi.org/10.1145/2254556.2254572

Enderle, S. (2019). *Topic modeling tool*. https://github.com/senderle/topic-modeling-tool.

Goldstone, A., & Underwood, T. (2014a). *DFR-browser*. https://github.com/agoldst/dfr-browser.

Goldstone, A., & Underwood, T. (2014b). The quiet transformations of literary studies: What thirteen thousand scholars could tell us. *New Literary History*, *45*(3), 359–384. https://doi.org/10.1353/nlh.2014.0025

Grootendorst, M. (2022). *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*. https://arxiv.org/abs/2203.05794

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. *Proceedings of the 7th Python in Science Conference (SciPy2008)*, 11–15. https://doi.org/10.25080/tcwv9851

Jockers, M. L. (2013). *Macroanalysis: Digital methods and literary history*. University of Illinois Press.

Kardos, M., Enevoldsen, K. C., & Nielbo, K. L. (2025). *topicwizard – a modern, model-agnostic framework for topic model visualization and interpretation*. https://arxiv.org/abs/2505.13034

McCallum, A. K. (2002). *MALLET: A machine learning for language toolkit*. http://mallet.cs.umass.edu/.

Mimno, D. (n.d.). *JsLDA: In-browser topic modeling*. https://github.com/mimno/jsLDA.

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50.

Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. In J. Chuang, S. Green, M. Hearst, J. Heer, & P. Koehn (Eds.), *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces* (pp. 63–70). Association for Computational Linguistics. https://doi.org/10.3115/v1/W14-3110

Sinclair, S., & Rockwell, G. (2016). *Voyant Tools*. https://voyant-tools.org/.

Waskom, M. L. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021