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.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Abhishek Tiwari

Reviewers:

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

Submitted: 22 April 2025
Published: unpublished

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.

Statement of Need

There are few interpretive modelling tools for bengali, and even fewer that support pedagogical engagement. Most LDA frameworks assume technical fluency, rely on notebooks or scripts, and give limited attention to interpretability. While backend libraries like Gensim (Řehůřek & Sojka, 2010) and Mallet are robust, they do not help users understand what a topic is, how it changes across documents, or how it connects to themes or questions.

Most open-source topic modelling tools prioritise scale or coherence evaluation. *pyLDAvis* (Sievert & Shirley, 2014), for instance, provides excellent topic distance visualisations and relevance-based ranking, but relies on pre-tokenised, preprocessed input and assumes English-language conventions. It offers no support for Indic scripts, and its interface—though informative—is not designed for novice users or pedagogical contexts. Similarly, Voyant Tools (Sinclair & Rockwell, 2016) provides a visually rich environment for text exploration, but lacks custom LDA configurability and does not support Bengali tokenisation or lemmatisation.

anvay lets users upload their own corpora, adjust parameters, and explore topic boundaries through interactive, multi-modal views. It has already been used in classroom and workshop settings to teach interpretive topic modelling to humanities students with no programming background.

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, 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)
  - Bar and pie charts for topic-document relations
  - Topic-word network graphs (NetworkX)
- **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 - **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/vinayak-dasgupta/anvay
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

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

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). pyLDAvis: Interactive topic model visualization. *ICML Workshop on Topic Models*.

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