Introducing tsLDA: A Workflow-Oriented Topic Modeling Tool

Simon Babb[1], Mia Celeste[2], Dana Harris[3], Ingrid Wu[2], Theo Bayard de Volo[4], Alfredo Gomez[2], Tatsuki Kuze[2], Taeyun Lee[2], David Mimno[5], and Alexandra Schofield[2]

[1]Haverford College (sbabb@haverford.edu)
[2]Harvey Mudd College (mia@miaceleste.dev,
[iwu,agomez,tkuze,talee]@hmc.edu, xanda@cs.hmc.edu)
[3]Claremont McKenna College (fteves22@cmc.edu)
[4]Pitzer College (tbayard@students.pitzer.edu)
[5]Cornell University (mimno@cornell.edu)

1 Motivation

Probabilistic topic models allow researchers to extract insights from large text corpora without labels or annotations (Blei, 2012). Existing topic modeling tools, however, require technical knowledge outside the expertise of many potential users. We introduce `tsLDA`, a publicly-available, open-source online topic modeling tool that is oriented towards non-expert users. Our contributions include a pre-development survey, the `tsLDA` tool itself, and a subsequent user study.

2 System

The basis of our work is `jsLDA`, a d3 web app developed by David Mimno to teach about probabilistic topic models (Mimno, 2020). `jsLDA` implements Gibbs sampling inference for latent Dirichlet allocation (Griffiths and Steyvers, 2004) to best explain how words in the corpus show up together. The algorithm iteratively determines from the corpus a probable set of *topics*, defined as probability distributions over words in a vocabulary. While popular existing software such as the command-line tool MALLET (McCallum, 2002) and the visualizer LDAVis (Sievert, 2020) require installation, `jsLDA` runs in the browser and is accessible to users unfamiliar with such tools.

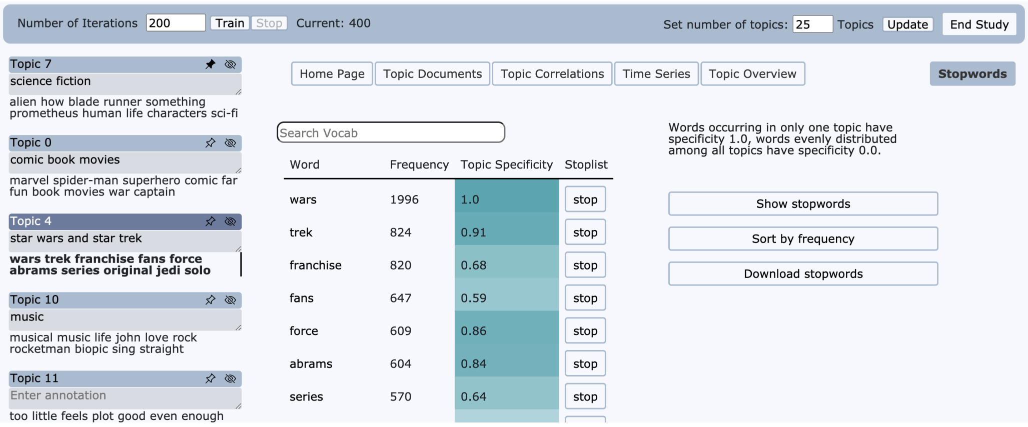
We refactored the `jsLDA` codebase to use React and Typescript (hence `tsLDA`). Responding to issues shared among topic modeling users we interviewed, we added functionality across the following categories: algorithm, visualization, and workflow. First, we implemented hyperparameter optimization, which can simply be selected before training to allow the inference of topics with asymmetric priors, i.e., different expectations of their probability. This reduces the likelihood of duplicate or multiple-subject topics, two common phenomena that confuse users (Wallach et al., 2009).

Next, users sought straightforward visualizations of trained models without writing their own plotting code. We added treemaps, which are space-filling plots for hierarchical data, to illustrate document-topic and topic-word distributions (Shneiderman, 1992). The proportions of topics within a document or words within a topic can be visualized while emphasizing the inherent hierarchical relationships stemming from the distributions. We also added the capacity for plots using categorical document-level data, such as authors or locations. These plots show the average topic score for a selected topic across all values in a categorical data field, allowing users to examine the relationship between topics and document metadata.

Finally, users wanted streamlined workflows to access necessary information to make sense of individual topics. In `tsLDA`, users can label topics with annotations, which are then used to represent the topic throughout the tool. The Topic Documents page of `tsLDA` supports highlighting salient words, which allows users to select a topic and visually measure how distinct each token in a document is to that topic (Chuang et al., 2012). We also created the Topic Overview page to let users explore a single topic in depth, which lists the most probable words and the most correlated topics.

3 Evaluation

To evaluate our tool and the functionality we introduced, we conducted an interactive user study. We provided fifteen participants with a version of the tool loaded with movie reviews from Rotten Tomatoes (Leone, 2020). Participants ranged from college students with programming experience but relatively little NLP experience to more senior scholars outside computing who work with large text collections. After a brief introduction to the software, we asked users to configure, train, and interact with



Topic 7
science fiction
alien how blade runner something prometheus human life characters sci-fi

Topic 0
comic book movies
marvel spider-man superhero comic far fun book movies war captain

Topic 4
star wars and star trek
wars trek franchise fans force abrams series original jedi solo

Topic 10
music
musical music life john love rock rocketman biopic sing straight

Topic 11
Enter annotation
too little feels plot good even enough

Words occurring in only one topic have specificity 1.0, words evenly distributed among all topics have specificity 0.0.

Word	Frequency	Topic Specificity	Stoplist
wars	1996	1.0	stop
trek	824	0.91	stop
franchise	820	0.68	stop
fans	647	0.59	stop
force	609	0.86	stop
abrams	604	0.84	stop
series	570	0.64	stop

Figure 1: A view of the stopword selector screen on tsLDA for a model trained for 400 iterations. Some topics on the left side have been annotated with tentative labels, such as "comic book movies."

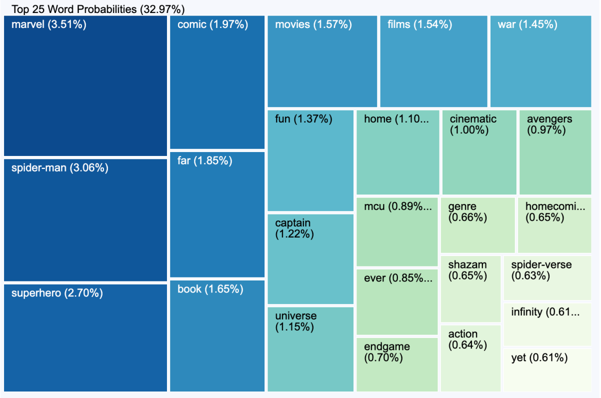


Figure 2: An example treemap showing relative probabilities of top words of a comic book topic.

a topic model to answer some analysis questions about the dataset, such as "What are some examples of movies in the dataset that overlap between action and comedy?" This allowed us to observe how participants used features of tsLDA to extract insights from text. We then interviewed them about their experience interacting with the tool. From the feedback we received, we isolated three common themes to guide future development.

Topic refinement. Our participants often determined parameters of the model, such as the number of topics, training iterations, and stopwords, with a trial and error process. After configuring and training a model, participants—including those new to topic models—inspected the resulting topics to assess their quality before deciding to train further, reset, alter the stopwords list, or continue their analysis with the existing model. This is an exciting result, as encouraging this iterative process is a goal of this tool. Our past expert interviews revealed that workflows involving topic models included significant time on trial and error for this process that ultimately relied on inspecting topics directly to make useful decisions.

Visualization. When inspecting topics, users generally found visualizations to be especially helpful. This included some original functionality of jsLDA, such as the dynamic stopword selector that emphasizes topic specificity (Figure 1), as well as time series views of topics. Users also were drawn to the newer functionality on the Topic Overview page, including extended top word lists and treemaps of word probabilities (Figure 2). However, users sometimes struggled to configure the visualizations to show the information they wanted. A way to sort stopwords by topic specificity or a more intuitive interface to smooth time series plots would be helpful.

Topic Comparison. Users wanted a better way to select and compare a subset of topics. While our tool offers a correlation visualization of all topics' correlations with each other, this figure was difficult to interpret for our users due to a lack of understanding of the pointwise mutual information metric and the large size of the matrix when there is a higher number of topics.

4 Next Steps

After conducting our user study, we are rethinking how to make functionality in tsLDA clearer by not only implementing the features above, but also embedding help information throughout the app. As requested by participants, we will also create a tutorial blog post and videos to demonstrate common tasks using tsLDA.

Acknowledgments. This work was supported by Harvey Mudd College internal grants (including the Vandiver, Class of ‘84, and Fletcher Jones Funds) and the National Science Foundation REU #1950885.

References

David M. Blei. 2012. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Jason Chuang, Christopher D. Manning, and Jeffrey Heer. 2012. Termite: visualization techniques for assessing textual topic models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI ’12, pages 74–77, Capri Island, Italy. Association for Computing Machinery.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235. Publisher: National Academy of Sciences Section: Colloquium.

Stefano Leone. 2020. Rotten tomatoes movies and critic reviews dataset. Available at: `https://www.kaggle.com/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset`.

Andrew Kachites McCallum. 2002. MALLET: A MAchine Learning for LanguagE Toolkit. `http://mallet.cs.umass.edu`.

David Mimno. 2020. mimno/jsLDA. Original-date: 2013-04-23T14:37:17Z.

Ben Shneiderman. 1992. Tree visualization with treemaps: 2-d space-filling approach. *ACM Transactions on Graphics (TOG)*, 11(1):92–99.

Carson Sievert. 2020. cpsievert/LDAvis. Original-date: 2014-03-05T06:17:16Z.

Hanna M Wallach, David M Mimno, and Andrew McCallum. 2009. Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems*, volume 22, pages 1973–1981. Curran Associates, Inc.