Trellis: Visual Topic Aggregation

Thomas Frederick Schaffner

A Master's Thesis

Presented to the Faculty

of Princeton University

in Candidacy for the Degree

of Master of Science in Engineering

Recommended for Acceptance

by the Department of

Computer Science

Adviser: Professor Barbara Engelhardt

June 2018

Abstract

With increasing availability of large volumes of text data, researchers from many fields have more documents than they can easily analyze by hand. Topic models are powerful computational tools that can be of great benefit to these researchers, but technical requirements can make these models inaccessible to a broad audience. With the goal of empowering users to examine their text data and refine models of their corpora based on domain expertise, Trellis is a visual tool for topic model curation and dataset exploration. By allowing researchers to visually interact with their datasets and aggregate topics within the model, Trellis enables an iterative process of adjusting a working hierarchical topic model and examining the corresponding dataset based on that working model.

Acknowledgements

I would like to thank professors Barbara Engelhardt and Brandon Stewart for their time, support, and advice throughout this process. I would also like to thank Allison Chaney for her advice and help, and I would like to thank both Gregory Gundersen and Allison Chaney for their work on Trellis.

I also thank Matthew Heinrich, Christopher Galea, and Oluwatosin Adewale for their feedback while I was preparing this thesis.

Finally I would like to thank my family for their continued love and support.

Contents

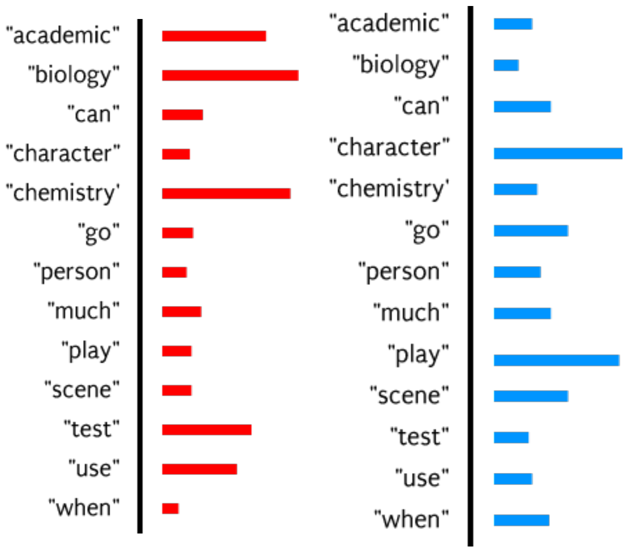
1 Introduction

Plain text documents are a valuable resource for researchers of many fields. Social sciences routinely make use of text data for exploratory work and both qualitative and quantitative analysis. The quantity and accessibility of research-usable text data is growing quickly as new documents are created digitally or old documents are digitized. This availability presents new opportunities and challenges for researchers. While it may be easier than in the past to find relevant documents for any given research problem, the number of documents can be too large to parse or analyze by hand. Consequently, researchers in many fields that traditionally do not require computational expertise are adopting tools and models from computer science. Specifically, topic models are powerful computational models of text corpora, representing corpus-specific "topics" and associating documents with these topics in different proportions.

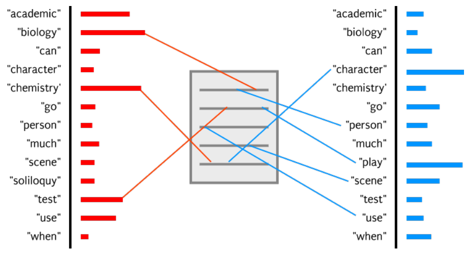
In many simple topic models, such as Latent Dirichlet Allocation (LDA) [3, 1], topics are defined as distributions over the vocabulary terms within a text corpus. Each individual topic essentially places a weight on all vocabulary terms. This distribution can be nearly uniform, placing similar weights on many terms, or it can highly weight only a few terms. It is also possible for the distributions of two topics to be nearly identical; there is no requirement that topics' highly-weighted terms be disjoint. Figure 1a shows two topic distributions over a small sample vocabulary.

An individual document is modeled as an unordered set of vocabulary terms. In these simple topic models, each word is associated with an individual topic (see Figure 1b). Across different documents, the proportions of words associated with each topic varies, as shown in Figure 1c. These proportions can be described as distributions over all topics, corresponding to different weightings for each document-topic pair.

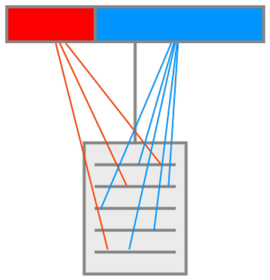
Trellis makes use of both of these kinds of distribution. Topic-vocabulary distributions are useful for clustering topics within a hierarchy and identifying highly weighted terms for aggregate topics. Trellis similarly uses the document-topic distributions when sorting documents by topic relevance. Section 4 discusses these features in more detail.



(a) Example topic distributions



(b) Word associations with topics



(c) A document's topic distribution

Figure 1: Topic-vocabulary and document-topic distributions for topic simple topic models

Topic models can be useful when exploring a dataset, providing organization to the documents of the text corpus. Additionally, the models can be used as tools to help measure properties of large datasets. For example, Amy Catalinac uses a topic model trained on thousands of Japanese political manifestos to measure the degree to which individual manifestos have to do with certain topics like national security [catalinac2016pork].

However, topic models are not trivial to use. The simplest and most straightforward models are dependent on parametrization. Choosing the "correct" number of topics for a model like LDA can be difficult. Additionally, validation of trained topic models is difficult, and can require significant additional work [5].

These usage difficulties can be a barrier to entry, especially for researchers without a formal background in probabilistic modeling. With increasing availability of text data for social science research, topic modeling has growing potential to be a widely applied method across many disciplines. It is therefore important to create new topic modeling tools with accessibility and general applicability in mind.

I here detail a visualization tool designed to help researchers explore topic models and text corpora while improving accessibility to those without significant backgrounds in topic modeling. Trellis allows users to aggregate the topics from a trained model in order to create a hierarchical topic model. Trellis uses multiple visualizations to represent the hierarchical model and provides controls for exploring full text documents.

2 Design Goals

The goal of Trellis is to empower users to explore and analyze text corpus data by improving accessibility and flexibility of topic models. A simple, trained topic model requires work and time to interpret. Trellis primarily provides interactive visualizations for organizing the topics from trained topic models, and additionally facilitates exploration of the underlying text data.

Trellis is designed to make use of topic models trained with large numbers of topics. Models with many topics may be difficult to work with by themselves, but their topics are more likely to be specific or meaningful. For a target user base of researchers, Trellis assumes that users have a rough mental model of the underlying topic structure for their text corpus. Or, at the very least, they have a rough idea regarding what constitutes valid or invalid models of the underlying topic structure. Trellis is therefore designed to allow and assist with aggregation of the topics provided by the user into a larger hierarchy. This allows the researcher to use the computational topic model while injecting their own expertise-informed mental model into the process.

Additionally, Trellis streamlines document exploration based on a topic model. By providing readily available access to relevant documents for each topic or aggregation of topics, the tool allows users to perform a range of tasks from exploring their corpus for informative texts to evaluating individual topics.

3 Related Work

3.1 Hierarchical Modeling

Hierarchical organization of a model's topics is not a new concept. Hierarchical Dirichlet Processes (HDPs) [17] and nested Chinese Restaurant Processes (nCRPs) [2], for example, make significant use of hierarchical structure. These models are useful and nonparametric approaches to determining cluster numbers can simplify the process of training a topic model. However, even these hierarchical models require further work to allow users to modify the hierarchy structure or inject their domain expertise. While it is possible to manually adjust the results of a topic model (hierarchical or not), it would likely require custom software and a strong understanding of the underlying model. Overall, this is not an easily accessible option for researchers without a background in probabilistic modeling.

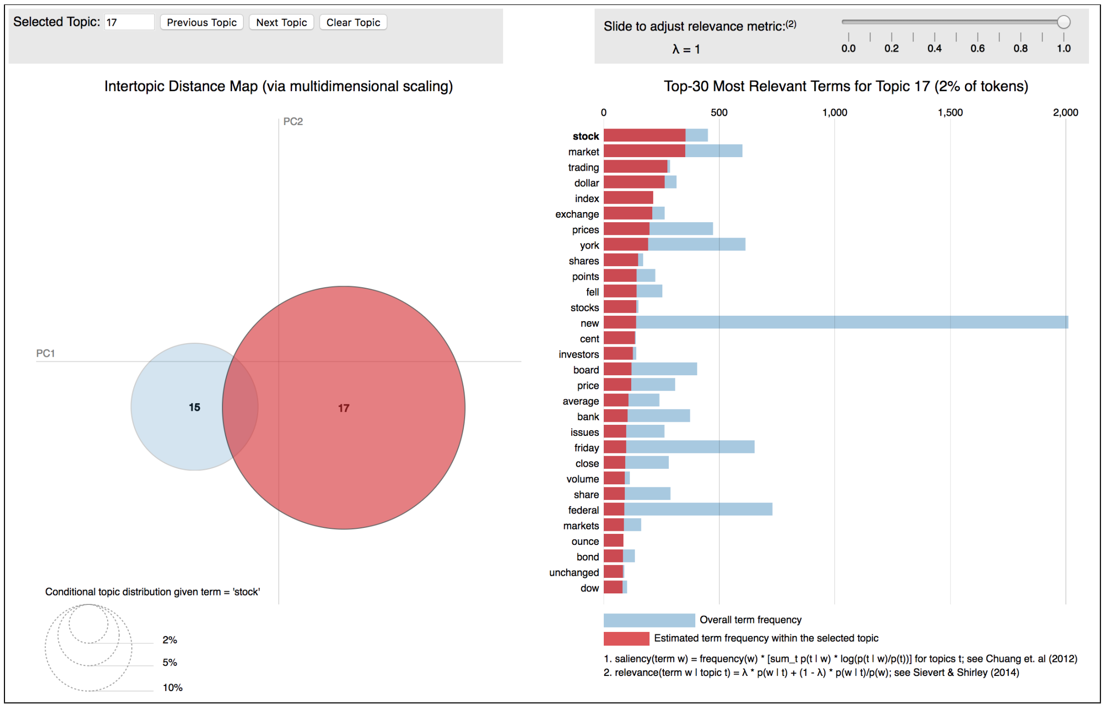
3.2 Model Visualization

Similarly, various tools exist to visualize and explore certain topic models. LDAvis [14] is a common package which allows users to explore topic models. LDAvis creates an interactive visualization of LDA models, displaying a 2-dimensional representation of the dissimilarities between topics. The tool lets users examine the vocabulary weights of each topic. Users can additionally aggregate topics into a single level of clusters, whose vocabulary distributions can be examined in the same manner as an individual topic.

Termite [8] aids in a more document-focused exploration of datasets, allowing users to examine topic-specific vocabulary distributions and full text documents within the tool itself. Termite also uses measures of "saliency" and "distinctiveness" [8] to order the representation of vocabulary term weightings per topic, with the goal of helping users more easily parse and explore the text corpus. Neither LDAvis nor Termite works with hierarchical topic models, though.

Alternatively, Hiérarchie [16] operates on hierarchical topic models but does not allow for as much exploration as either LDAvis or Termite. Hiérarchie uses a sunburst visualization to represent a hierarchical organization of topics. Users can then interact with the sunburst to examine the top-weighted vocabulary terms in any topic or zoom in to an intermediate-level or leaf-level topic in the hierarchy. Hiérarchie represents a hierarchical structure well, but has limited value as an exploratory tool as users cannot read the original documents within the tool.

Additional tools exist as well, like TopicFlow [10], Topic Explorer [11], other visual tools, and custom dataset-specific scripting by experienced researchers. This is not intended as an exhaustive list of topic model visualization tools, but rather serves to highlight some common areas of weakness within the ecosystem of visualization tools.

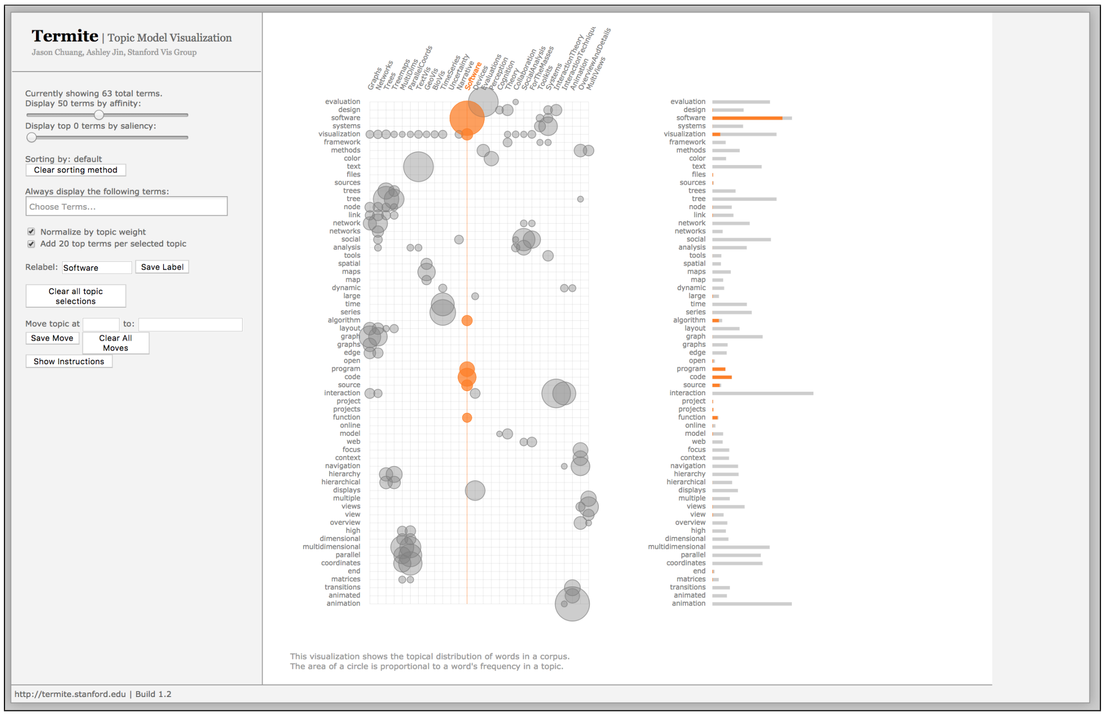


1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 2: Screenshot of LDAvis interface from R package implementation [14]

3.3 Weaknesses of Current Tools

Though the tools shown here have varying degrees of interactivity or complexity to their models and interfaces, the underlying data models are generally static. The tools are often useful for evaluating a model or exploring a dataset given a certain model, but cannot be used by domain experts to quickly adjust or correct a topic model. In particular, visual


Termite | Topic Model Visualization
Jason Chuang, Ashley Jin, Stanford Vis Group

Currently showing 63 total terms.
Display 50 terms by affinity:
Display top 0 terms by saliency:

Sorting by: default

Always display the following terms:

☑ Normalize by topic weight
☑ Add 20 top terms per selected topic

Relabel: Software

Move topic at to:

This visualization shows the topical distribution of words in a corpus.
The area of a circle is proportional to a word's frequency in a topic.

http://termite.stanford.edu | Build 1.2

Figure 3: Screenshot of Termite [8] interface from web demo [7]

tools for modifying hierarchical models or constructing aggregate models are particularly lacking. Therefore, though some tools support model-based exploration of documents, common tools do not support in-tool exploration of documents based on a complex model that is adjustable by the user.

Researchers can of course create their own customized software to perform these functions. However, a single program written to explore a specific dataset or to answer a specific research question is not likely to generalize to novel datasets or problems. More importantly, creating custom programs for topic model exploration or modification requires significant technical experience. These limitations mean that custom programs, while useful, are unlikely to be accessible to the wider audience of researchers is social sciences.

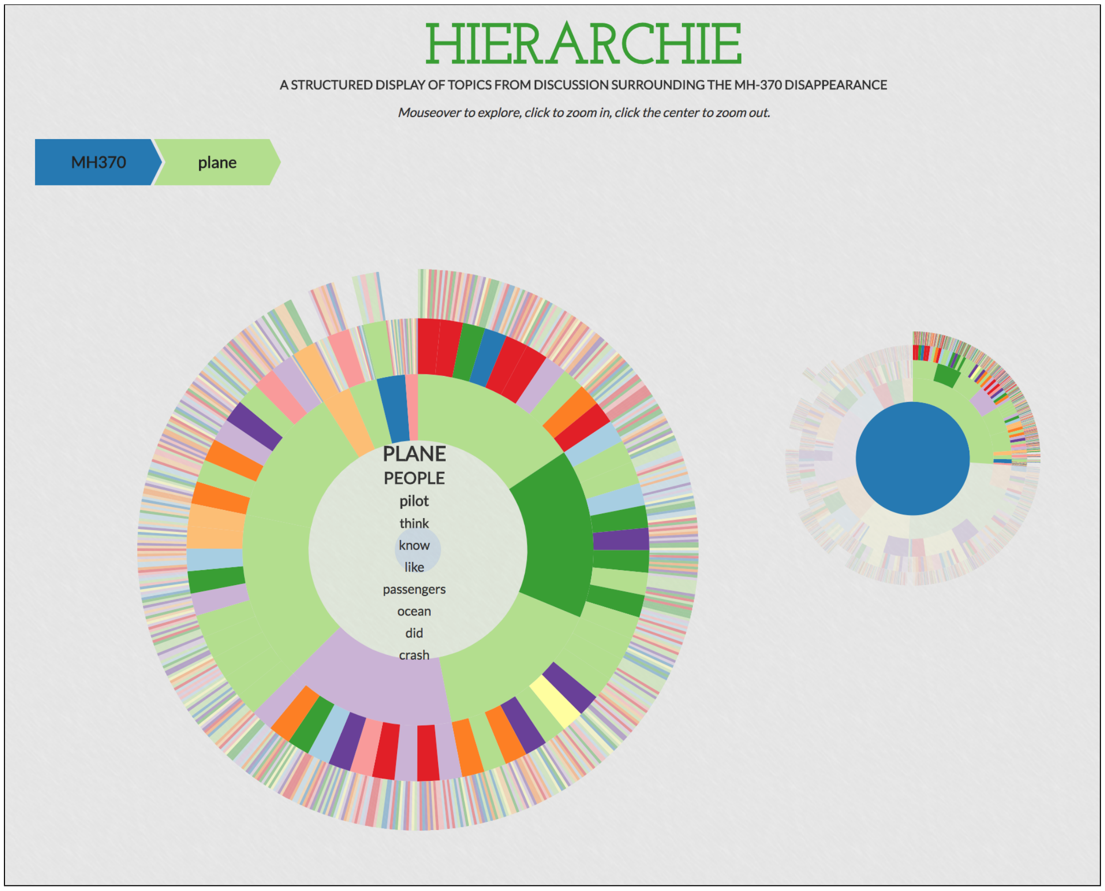
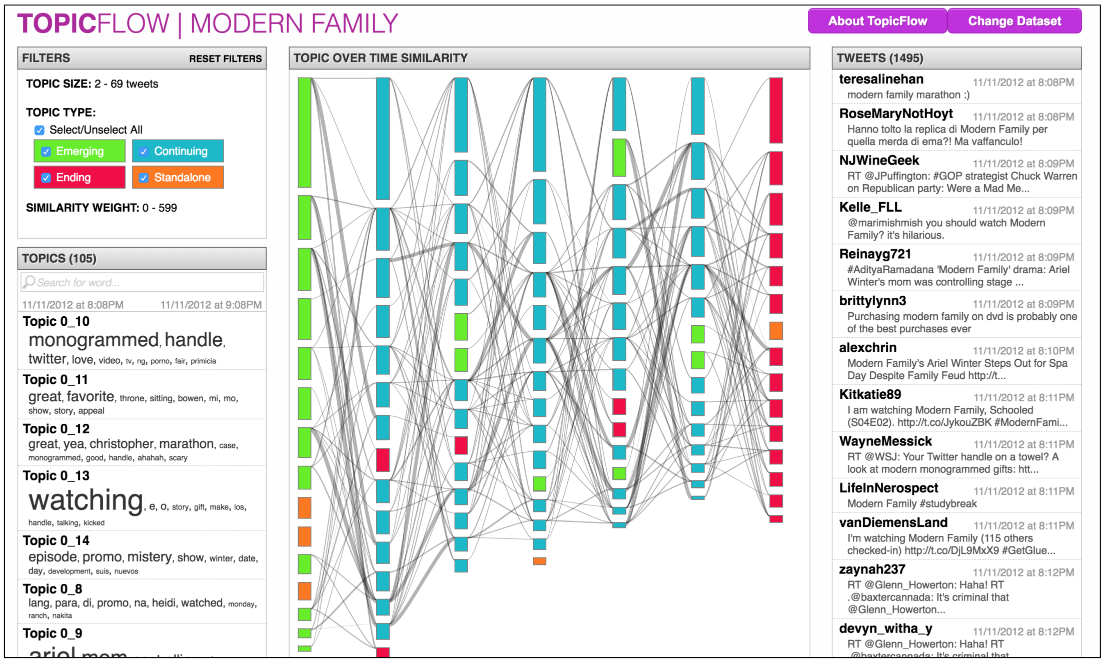

HIERARCHIE
A STRUCTURED DISPLAY OF TOPICS FROM DISCUSSION SURROUNDING THE MH-370 DISAPPEARANCE
Mouseover to explore, click to zoom in, click the center to zoom out.
MH370
plane
PLANE
PEOPLE
pilot
think
know
like
passengers
ocean
did
crash

Figure 4: Screenshot of the Hiérarchie [16] web version [15]



FILTERS RESET FILTERS

TOPIC SIZE: 2 - 69 tweets

TOPIC TYPE:

☑ Select/Unselect All

☑ Emerging ☑ Continuing

☑ Ending ☑ Standalone

SIMILARITY WEIGHT: 0 - 599

TOPICS (105)

11/11/2012 at 8:08PM 11/11/2012 at 9:08PM

Topic 0_10
monogrammed, handle, twitter, love, video, tv, ng, porno, fair, primicia

Topic 0_11
great, favorite, throne, sitting, bowen, mi, mo, show, story, appeal

Topic 0_12
great, yea, christopher, marathon, case, monogrammed, good, handle, ahahah, scary

Topic 0_13
watching, e, o, story, gift, make, los, handle, talking, kicked

Topic 0_14
episode, promo, mistery, show, winter, date, day, development, suis, nuevos

Topic 0_8
lang, para, di, promo, na, heidi, watched, monday, ranch, nakita

Topic 0_9

TWEETS (1495)

teresalinehan 11/11/2012 at 8:08PM
modern family marathon :)

RoseMaryNotHoyt 11/11/2012 at 8:08PM
Hanno tolto la replica di Modern Family per quella merda di ema?! Ma vaffanculo!

NJWineGeek 11/11/2012 at 8:09PM
RT @JPuffington: #GOP strategist Chuck Warren on Republican party: Were a Mad Me...

Kelle_FLL 11/11/2012 at 8:09PM
@marimishmish you should watch Modern Family? it's hilarious.

Reinayg721 11/11/2012 at 8:09PM
#AdityaRamadana 'Modern Family' drama: Ariel Winter's mom was controlling stage ...

brittylynn3 11/11/2012 at 8:09PM
Purchasing modern family on dvd is probably one of the best purchases ever

alexchrin 11/11/2012 at 8:10PM
Modern Family's Ariel Winter Steps Out for Spa Day Despite Family Feud http://t...

Kitkatie89 11/11/2012 at 8:11PM
I am watching Modern Family, Schooled (S04E02). http://t.co/JykouZBK #ModernFami...

WayneMessick 11/11/2012 at 8:11PM
RT @WSJ: Your Twitter handle on a towel? A look at modern monogrammed gifts: htt...

LifeInNerospect 11/11/2012 at 8:11PM
Modern Family #studybreak

vanDiemensLand 11/11/2012 at 8:11PM
I'm watching Modern Family (115 others checked-in) http://t.co/DjL9MxX9 #GetGlue...

zaynah237 11/11/2012 at 8:12PM
RT @Glenn_Howerton: Haha! RT .@baxtercannada: It's criminal that @Glenn_Howerton...

devyn_witha_y 11/11/2012 at 8:12PM
RT @Glenn_Howerton: Haha! RT

Figure 5: Screenshot of the TopicFlow [10] web interface [9]

4 Trellis

4.1 Functionality Goals

In line with the design goals listed in Section 2, I had several main functionality goals when designing and building Trellis.

Interaction with hierarchical structure I designed Trellis to allow users to construct and modify the structure of a hierarchical organization of topics. By providing an intuitive visual interface for construction of hierarchies, Trellis helps users better align computationally-generated topic models with their mental models shaped by domain expertise.

Exploration of underlying data For Trellis to be a useful data exploration tool, it needs to expose the underlying text data to its users. To that end, Trellis can sort documents by any topic or aggregation of topics. These documents can be read in full within Trellis, allowing users to explore the text corpus without leaving the tool.

4.2 Major Features

The interface for Trellis consists of two pages. A startup page houses initialization options while the main page has a main view for hierarchy visualizations and a sidebar for controls and information. Controls on the sidebar select which of the main visualizations to display in the main view and which additional controls or information to display in the sidebar. Trellis can also export SVG graphics, save the current working hierarchical model (for

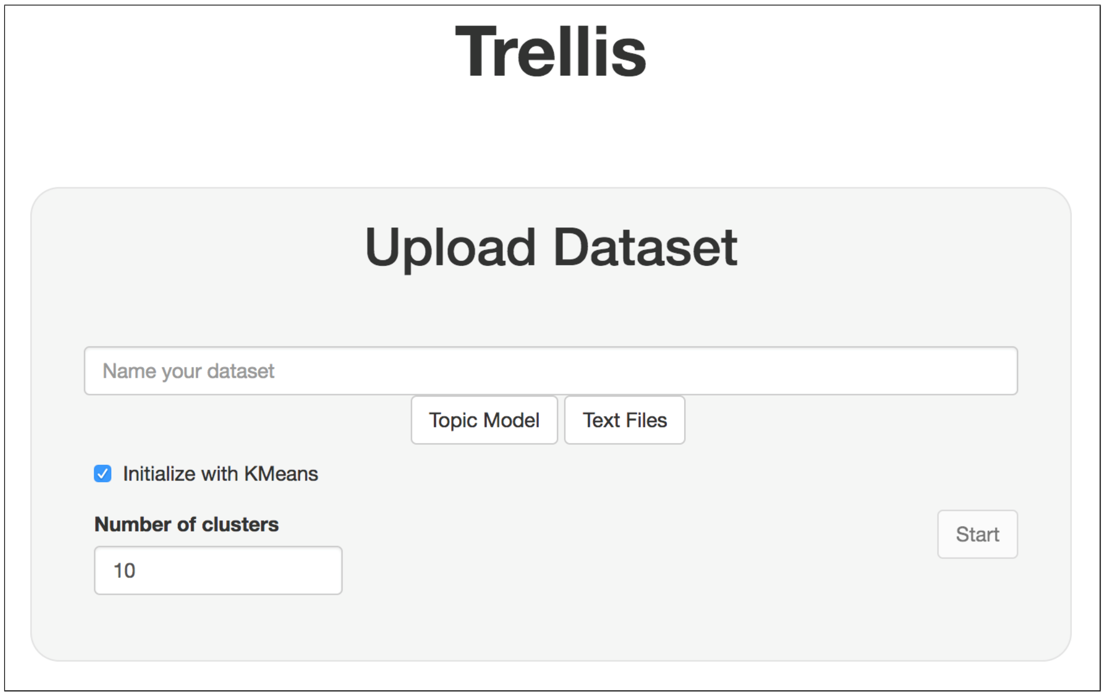


Figure 6: Trellis Initialization Pane

future use with the tool), and display text documents from the corpus if linked by the user. An instruction pane additionally helps to orient and inform new users.

Initialization Panel When first opened, Trellis starts with a single panel visible (Shown in Figure 6). This panel contains selectors for a `.RData` model file and optionally a directory containing all text files in the corpus. If the directory is provided, users will be able to open text files within the tool. The initialization panel also contains an option for performing a preliminary topic clustering. If selected, Trellis will cluster all topics in the provided `.RData` file before displaying either main visualization. Otherwise, Trellis will start with all provided topics as children of a single root node.

Hierarchy Visualizations These visualizations convey the organizational structure of the user's working topic model and support modifications to the hierarchy itself. Trellis has two main visualizations: a “Bubble View” based on a circle-packing hierarchy representation and a “Tree View” based on a simple tree layout. Both visualizations are built on top of `d3.js` [4] and software originally created by Gregory Gundersen.

In both visualizations, the user’s working topic model is represented as a hierarchy of nodes. Each node corresponds to a topic, where each topic is either from the original computational model or some aggregation of other topics. Consequently, all original topics (from the originally trained model) correspond to leaf nodes in the hierarchy. All non-leaf nodes are therefore user-generated or user-curated aggregations. Note that Trellis does not enforce uniform depth for all leaf nodes.

The Bubble View (Figure 7a) emphasizes the leaf nodes of a user’s working hierarchical topic model. All leaf nodes are always displayed and visually distinct from aggregate nodes. Graphical groupings of leaf nodes encode clusters of topics in the hierarchical model, and can be useful as an overview of the constituent “original” topics within any aggregate topic. The Bubble View is designed as an overview of the distribution of leaf nodes, giving users a broad sense of the working model and how it agrees or disagrees with their own mental models.

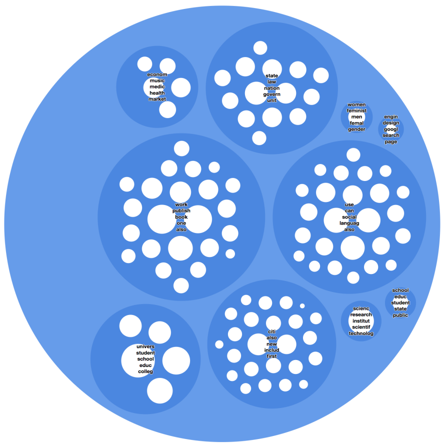
The Tree View (Figure 7b) alternately emphasizes the organizational structure of the hierarchy. Leaf nodes are not always visible in the Tree View, as a key functional difference allows users to collapse nodes of the tree and hide their descendants. When left expanded, the Tree View represents tree width and height in a more easily parsable manner than the Bubble View. Users can then collapse nodes in order to focus on individual branches of the hierarchical model. Together, the Bubble View and Tree View are designed to help users compare working models with their mental models while also providing detailed information for exploration and modification of the working topic structure.

Both the Bubble View and Tree View are interactive visualizations with controls for modifying the underlying hierarchy of nodes. Users can manually move or merge nodes with simple drag-and-drop actions. The specific outcome of a dragging action is determined by the source node (the node being dragged), the target node (the node at the end of the drag action), and the shift key as a modifier. Any node but the tree root is a valid source node, and any non-leaf node is a valid target. If the user drags a source node onto a leaf node, the leaf's immediate parent is selected as the target, allowing for easier dragging actions in the Bubble View.

If the source node is a leaf, the drag operation moves the source node. If the user holds shift while performing the action, Trellis creates a new, childless node as a direct descendant of the target node. The source node then becomes a direct descendant of the new intermediate node. Without the shift key, no new nodes are created and the source node becomes a child of the target.

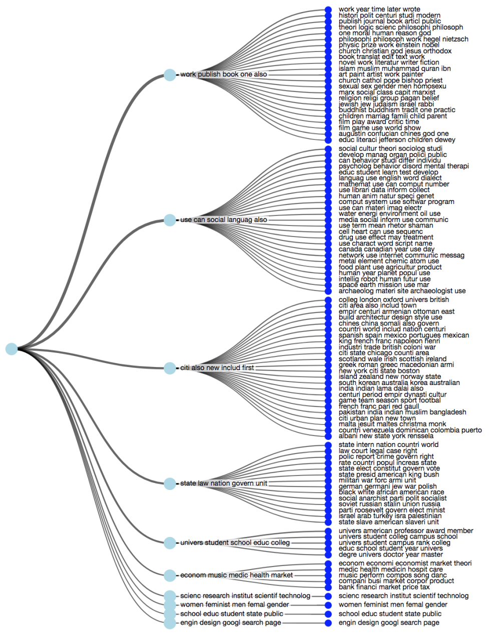
If the source node is an intermediate node, unmodified behavior is to merge both nodes. Specifically, if the user does not hold the shift key, the source node is deleted and all descendants of the source node become descendants of the target node. If the shift node is held, the source node remains intact, and becomes a child of the target node. Trellis removes all newly childless intermediate nodes after the move or merge operation is complete.

These four drag-and-drop cases produce a flexible system for manually adjusting the hierarchy structure of a working model. Individual nodes can be moved or combined, and users can create new levels in the hierarchy. Both visualizations also support zooming and panning to aid users adjusting or examining subsets of the working hierarchy. Finally, users can hover over nodes with the mouse to gain a preview of some extra information in the sidebar, or select nodes by clicking on them to enable further sidebar controls.



econom music medic health market
state law nation govern unit
women feminist men femal gender
engin design googl search page
work publish book one also
use can social languag also
school educ student state public
scienc research institut scientif technolog
univers student school educ colleg
citi also new includ first

(a) Nested circle visualization



work publish book one also
use can social languag also
citi also new includ first
state law nation govern unit
univers student school educ colleg
econom music medic health market
scienc research institut scientif technolog
women feminist men femal gender
school educ student state public
engin design googl search page
work year time later wrote
histori polit centuri studi modern
publish journal book articl public
theori logic scienc philosophi philosoph
one moral human reason god
philosophi philosoph work hegel nietzsch
physic prize work einstein nobel
church christian god jesus orthodox
book translat edit text work
novel work literatur writer fiction
islam muslim muhammad quran ibn
art paint artist work painter
church cathol pope bishop priest
sexual sex gender men homosexu
marx social class capit marxist
religion religi group pagan belief
jewish jew judaism israel rabbi
buddhist buddhism tradit one practic
children marriag famili child parent
film play award critic time
film game use world show
augustin confucian chines god one
educ literaci jefferson children dewey
social cultur theori sociolog studi
develop manag organ polici public
can behavior studi differ individu
psycholog behavior disord mental therapi
educ student learn test develop
languag use english word dialect
mathemat use can comput number
use librari data inform collect
human anim natur speci genet
comput system use softwar program
use can materi imag electr
water energi environment oil use
media social inform use communic
use term mean rhetor shaman
cell heart can use sequenc
drug use effect may treatment
use charact word script name
canada canadian year use day
network use internet communic messag
metal element chemic atom use
food plant use agricultur product
human year planet popul use
intellig robot human futur use
space earth mission use mar
archaeolog materi site archaeologist use
colleg london oxford univers british
citi area also includ town
empir centuri armenian ottoman east
build architectur design style use
chines china somali also govern
countri world includ nation centuri
spanish spain mexico portugues mexican
king french franc napoleon henri
industri trade british coloni war
citi state chicago counti area
scotland wale irish scottish ireland
greek roman greec macedonian armi
new york citi state boston
island zealand new norway state
south korean australia korea australian
india indian lama dalai also
centuri period empir dynasti cultur
game team season sport footbal
french franc pari red gaull
pakistan india indian muslim bangladesh
citi urban plan new town
malta jesuit maltes christma monk
countri venezuela dominican colombia puerto
albani new state york renssela
state intern nation countri world
law court legal case right
polic report crime govern right
rate countri popul increas state
state elect constitut govern vote
state presid american king bush
militari war forc armi unit
german germani jew war polish
black white african american race
social anarchist parti polit socialist
soviet russian stalin union russia
parti roosevelt govern elect minist
israel arab turkey isra palestinian
state slave american slaveri unit
univers american professor award member
univers student colleg campus school
univers student campus rank colleg
educ school student year univers
degre univers doctor year master
econom economi economist market theori
medic health medicin hospit care
music perform compos song danc
compani busi market corpor product
bank financi market price tax
scienc research institut scientif technolog
women feminist men femal gender
school educ student state public
engin design googl search page

(b) Tree visualization

Figure 7: Trellis' two options for main hierarchy visualization

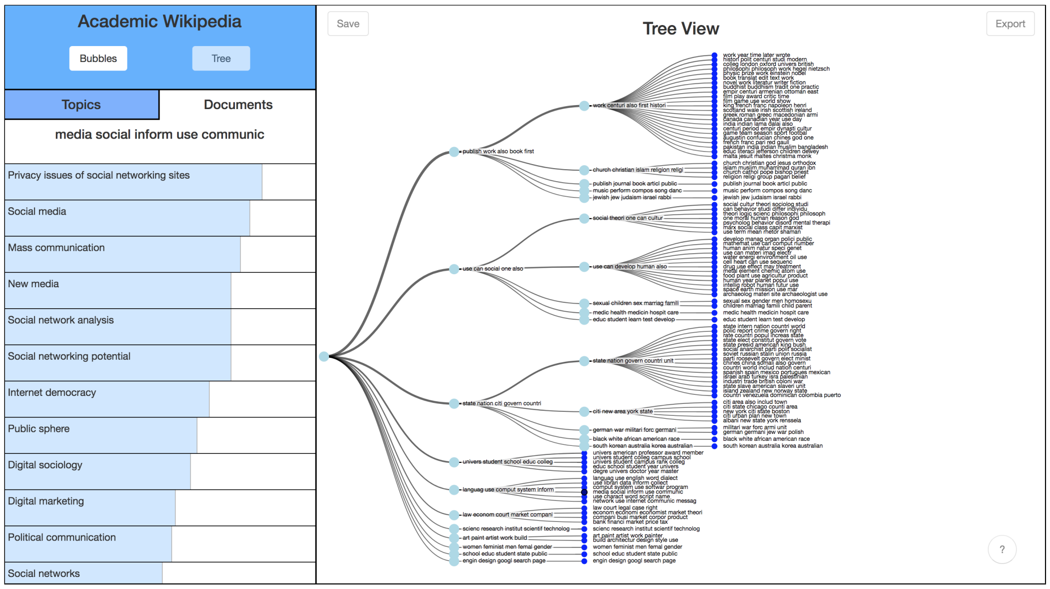
Sidebar The sidebar has several main components. The top of the sidebar displays the name of the current working model (given by the user on launch) and provides controls for selecting which main view is active and which sidebar content is active. Beyond these controls, the sidebar has two possible modes: Topic Controls and a Document List.

In addition to the interactions within the main visualizations, the sidebar Topic Controls provide several more methods of adjusting the working hierarchical model (shown in Figure 8b). After the user has selected a node in the main view, the sidebar shows options for renaming the selected node, deleting the selected node, and clustering the children of the selected node. Renaming a node does not adjust the hierarchy structure, but increases readability and facilitates communication of information about the model. Both other controls, though, are useful for quickly making structural modifications to the working model. Deleting a selected node reassigns all of the children of the selected node as children of the selected node's parent. Clustering the children of a node creates new nodes as children of the selected node, and assigns the original children of the selected node to the new nodes based on a k-means clustering. Trellis uses each topic's distribution over vocabulary terms as input to the k-means clustering.

As an alternative to the Topic Controls pane, the Document List is designed primarily as a means of gathering information (shown in Figure 8a). Whenever the user hovers over or selects a topic from the main view, the Document List shows a list of document titles from the text corpus sorted by the proportion of the document attributed to the selected topic. Each document name encodes its topic weighting as a proportionally filled bar. The document name also serves as a link to open the Document Viewer, so users can simply click a document name to open the full text within Trellis.

Document Viewer In order to enable users to fully explore their data sets, Trellis contains a Document Viewer (Shown in Figure 9). By clicking on a document from the Document List, users open the full text file as a popup window. If the user does not provide the relevant text files during initialization, the Document Viewer will remain blank.

Image Exporting With either main visualization active, users can export an SVG image of the current state of the active visualization. Clicking the Export button once toggles



Academic Wikipedia

Bubbles | Tree

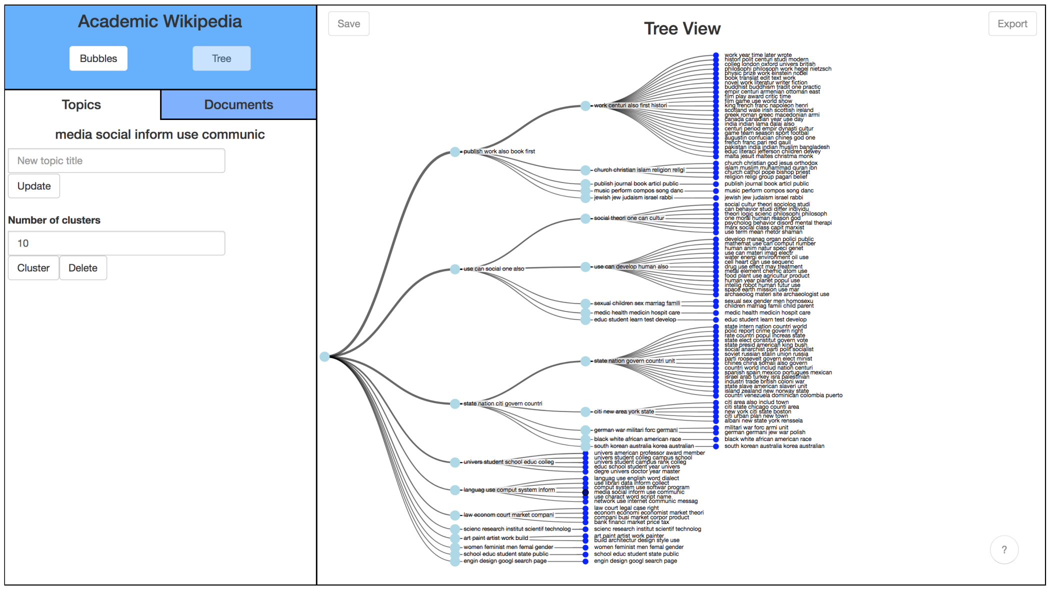
Topics | Documents

media social inform use communic

- Privacy issues of social networking sites
- Social media
- Mass communication
- New media
- Social network analysis
- Social networking potential
- Internet democracy
- Public sphere
- Digital sociology
- Digital marketing
- Political communication
- Social networks

Save
Tree View
Export
publish work also book first
work centuri also first histori
church christian islam religion religi
publish journal book articl public
music perform compos song danc
jewish jew judaism israel rabbi
use can social one also
social theori one can cultur
use can develop human also
sexual children sex marriag famili
medic health medicin hospit care
educ student learn test develop
state nation citi govern countri
state nation govern countri unit
citi new area york state
german war militari forc germani
black white african american race
south korean australia korea australian
univers student school educ colleg
languag use comput system inform
law econom court market compani
scienc research institut scientif technolog
art paint artist work build
women feminist men femal gender
school educ student state public
engin design googl search page
?

(a) Trellis layout with document list selected



Academic Wikipedia

Bubbles | Tree

Topics | Documents

media social inform use communic

New topic title

Update

Number of clusters

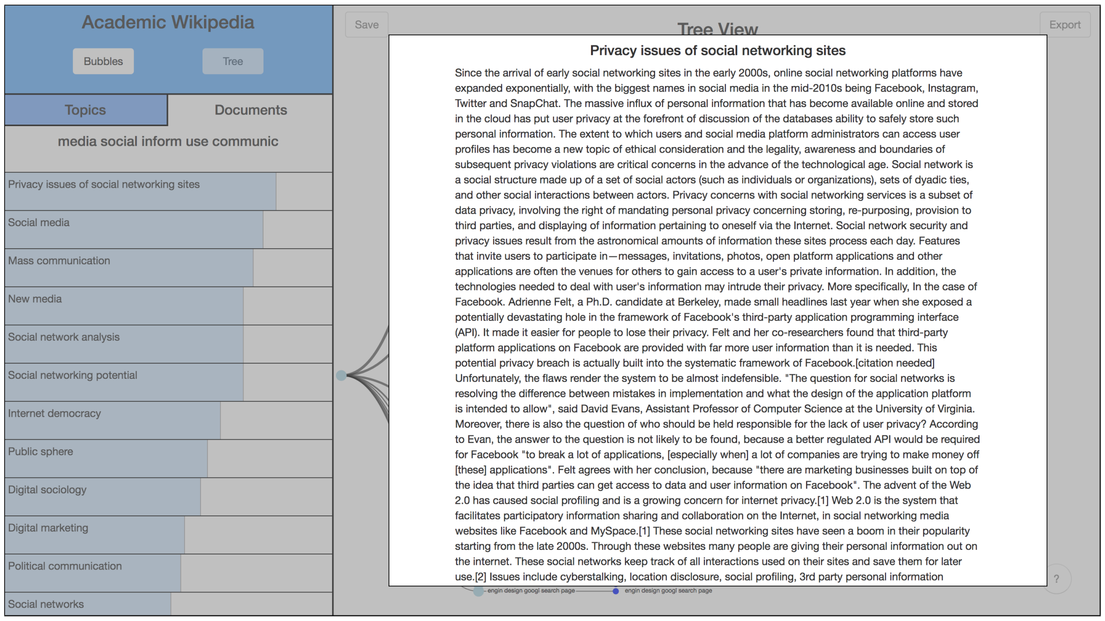
10

Cluster | Delete

Save
Tree View
Export
publish work also book first
work centuri also first histori
use can social one also
state nation citi govern countri
univers student school educ colleg
languag use comput system inform
law econom court market compani
?

(b) Trellis layout with topic controls selected

Figure 8: Two options for the Trellis sidebar



Privacy issues of social networking sites

Since the arrival of early social networking sites in the early 2000s, online social networking platforms have expanded exponentially, with the biggest names in social media in the mid-2010s being Facebook, Instagram, Twitter and SnapChat. The massive influx of personal information that has become available online and stored in the cloud has put user privacy at the forefront of discussion of the databases ability to safely store such personal information. The extent to which users and social media platform administrators can access user profiles has become a new topic of ethical consideration and the legality, awareness and boundaries of subsequent privacy violations are critical concerns in the advance of the technological age. Social network is a social structure made up of a set of social actors (such as individuals or organizations), sets of dyadic ties, and other social interactions between actors. Privacy concerns with social networking services is a subset of data privacy, involving the right of mandating personal privacy concerning storing, re-purposing, provision to third parties, and displaying of information pertaining to oneself via the Internet. Social network security and privacy issues result from the astronomical amounts of information these sites process each day. Features that invite users to participate in—messages, invitations, photos, open platform applications and other applications are often the venues for others to gain access to a user's private information. In addition, the technologies needed to deal with user's information may intrude their privacy. More specifically, In the case of Facebook. Adrienne Felt, a Ph.D. candidate at Berkeley, made small headlines last year when she exposed a potentially devastating hole in the framework of Facebook's third-party application programming interface (API). It made it easier for people to lose their privacy. Felt and her co-researchers found that third-party platform applications on Facebook are provided with far more user information than it is needed. This potential privacy breach is actually built into the systematic framework of Facebook.[citation needed] Unfortunately, the flaws render the system to be almost indefensible. "The question for social networks is resolving the difference between mistakes in implementation and what the design of the application platform is intended to allow", said David Evans, Assistant Professor of Computer Science at the University of Virginia. Moreover, there is also the question of who should be held responsible for the lack of user privacy? According to Evan, the answer to the question is not likely to be found, because a better regulated API would be required for Facebook "to break a lot of applications, [especially when] a lot of companies are trying to make money off [these] applications". Felt agrees with her conclusion, because "there are marketing businesses built on top of the idea that third parties can get access to data and user information on Facebook". The advent of the Web 2.0 has caused social profiling and is a growing concern for internet privacy.[1] Web 2.0 is the system that facilitates participatory information sharing and collaboration on the Internet, in social networking media websites like Facebook and MySpace.[1] These social networking sites have seen a boom in their popularity starting from the late 2000s. Through these websites many people are giving their personal information out on the internet. These social networks keep track of all interactions used on their sites and save them for later use.[2] Issues include cyberstalking, location disclosure, social profiling, 3rd party personal information

Figure 9: Full layout of Trellis with open document

Export Mode, which hides the sidebar. In Export Mode, users can still zoom, pan, and collapse nodes (in the Tree View). Users can then export an SVG image or exit Export Mode.

File Saving and Restoration Users can save the current state of their working models as `.RData` files. The save file includes all original model data, plus the structure of the hierarchy and any manually created topic titles. On startup, users can provide one of these saved files as the model file. Loading a saved file overrides any startup options, instead restoring the state of the saved file's working model.

4.3 Implementation

GitHub Repository: **`https://github.com/ajbc/topic-bubbles`**

Trellis is built as an RShiny [6] application, making use of `d3.js` [4] for the visualizations.

The back-end stores model state and handles file selection and saving. For efficiency, Trellis uses the `ShinyFiles` [12] package to handle file interactions. `ShinyFiles` uses R to explore local file systems, allowing the R server to simply load a file instead of uploading a copy of the file over a network to a locally hosted server. This speeds up the process of loading and saving files, but also means that Trellis only functions locally; Trellis cannot currently be deployed remotely.

Both main visualizations are implemented as `HTMLWidgets` [18] and based on software originally written by Gregory Gundersen. Each widget receives data from the R server and uses `d3.js` to render a visualization. Interactions native to a visualization (drag-and-drop, zooming, panning, and collapsing nodes) are handled within the `HTMLWidget`. Additional javascript code handles broader controls like selecting a visualization or exporting an SVG.

All software for Trellis is available on GitHub, including an example processing script that fits a structural topic model with the STM package [13] and saves the necessary data fields in the correct format for Trellis. Note that although this example script relies specifically on the data format of STM, any topic model with topic-vocabulary distributions and document-topic distributions can be reformatted to work with Trellis.

4.4 Current Usage

Trellis is currently in trial usage by two graduate students. Informal, preliminary feedback from one user indicates that the tool is useful for curating a topic model, though Trellis lacks some features present in other tools. This feedback specifically suggested the inclusion of output formatting options for compatibility with other tools, as discussed in Section 5.

5 Future Work

The currently-available software is fully functional, providing users with a tool for exploration of text corpora and construction of hierarchical topic models. However, several accessibility-related updates would be useful for researchers, and rigorous evaluation would be useful for identification of further areas of potential improvement.

5.1 Updates to Trellis

Though the software is currently available on a GitHub repository, the installation of Trellis requires manual downloading of the repository and installation of dependencies. By creating and publishing an R package, researchers will have a streamlined installation and update process.

Additionally, I want to provide improved support for a broader set of data formats. There are quite a few different kinds of topic model, many of which have their own R packages for training a model. The example processing script provided with Trellis is designed specifically for output from the STM R package. The the concepts involved in the tool are generally applicable, though. Additional processing scripts for reformatting other topic models would therefore be useful.

The output format of Trellis save files is also currently very specific. Given the large ecosystem of other visualization tools, additional output formatting options would be useful. Some researchers are already familiar and experienced with certain tools, and integration of Trellis with these existing tools would increase the ease of use and accessibility of Trellis. By allowing researchers to choose a cut of the hierarchy, Trellis would be able to export the new leaves as a flattened topic model. The specific data for the topic model could be formatted to work well with specific tools, such as LDAvis.

5.2 Evaluation

Current evaluation of Trellis consists of feedback from initial users. The aim of this evaluation is to discover software bugs and improve any major user interface hurdles, but it is informal. Additionally, based on the small number of users, it would be difficult to draw meaningful conclusions from this feedback. I intend to more rigorously evaluate Trellis in the future.

Primarily, I wish to examine the usefulness of Trellis with respect to research questions focusing on text data within social sciences. One approach will involve subjects' real-world research questions. I will select test subjects who are researchers in social sciences, with their own simple research questions where topic modeling is a useful tool. For the sake of availability, these subjects will likely be graduate students.

Subjects will be selected for one of two groups: novice or expert. The novice group would consist of researchers with rudimentary understanding of topic modeling, but without experience using topic modeling in the context of research. The expert group would then consist of researchers who have familiarity with at least one topic modeling tool and experience using topic modeling in the context of research.

Each subject will be set up with Trellis and given time to address their research question. We will then bring in each subject for an interview about the process of using Trellis, focusing on intuitiveness of the interface, availability of desired features, and effectiveness of available features. We will then provide each subject with a working installation of LDAvis. They will be given some time to explore the tool, after which they will be given a questionnaire comparing Trellis and LDAvis.

However, there are several challenges with this study. It will likely be difficult to find an adequate number of participants for the study to produce significant results, given that each participant must have at least some knowledge of topic modeling and some participants must be experienced with a topic modeling tool. Additionally, finding researchers with research questions of the appropriate scope will pose more of a challenge.

An alternate approach is to break down the evaluation of Trellis into multiple tests. Evaluations that are smaller in scope will be easier to coordinate and more feasible to run. However, a major proposed strength of Trellis is the interaction between model modifications and document exploration. While smaller, separate tests can provide evidence for the usability and usefulness of Trellis, the next two tests proposed here are not comprehensive and do not address the iterative nature of user interactions with the tool.

First, as a simple test for exploration functionality, I propose curating a text corpus and training a topic model such that there exist several "inter-topic" documents. Essentially, given a topic model with a large number of topics, the text corpus would contain several documents of interest that are not highly weighted on any of the individual topics. However, these documents of interest would be highly weighted given a certain underlying "ground truth" aggregation of the original topics.

Test subjects would be given the trained model, the text dataset, and a set of questions asking whether any documents relating to certain ideas are present in the dataset. For example, subjects could be given a dataset of academic documents and asked whether any documents were present relating to historical economics. This test would attempt to evaluate the ability of Trellis to support users searching for targeted documents or subjects. Model curation and aggregation may be helpful in such a task, but are not the primary focus.

Second, I propose asking test subjects to create aggregate models based on datasets related to their own fields of study. Keeping in mind the potentially limiting factor of subject availability, subjects would primarily be graduate students grouped by their field of research. All participants within the same group would receive the same curated dataset and original topic model, and asked to create an aggregate model. They would then be asked to fill out a simple questionnaire regarding several other participants' (anonymized) aggregate models. These questions would focus on the meaningfulness of aggregate topics, parsimony of the resulting hierarchy, and an overall scoring of the model.

As a modification to the study, users would be asked to generate flat topic models. These models could originate from a retraining on different numbers of topics, another tool's output, or a cut of a working model's hierarchy in Trellis. Subjects would be randomly assigned a tool to work with (either Trellis or LDAvis, for example). Their resulting model, and the tool used to create it, would again be anonymized before being graded by other participants. Because of the option of retraining a model, the technical background required for subjects of this modified study would need to be stronger than in the other tests. The problem of subject availability would therefore be even more restrictive in this scenario.

6 Conclusion

In conclusion, Trellis is a visualization tool for aggregating topic models and exploring text corpora. Functionality for modifying aggregate hierarchical topic models allows researchers to inject domain expertise into the modeling process. In-tool document display then enables users to both subjectively evaluate their working models and make use of those models to analyze their datasets. While Trellis would benefit from certain feature additions, the current software is functional and freely available.

References

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

[2] David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] Michael Bostock et al. D3. js. *Data Driven Documents*, 492:701, 2012.

[5] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296, 2009.

[6] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. *shiny: Web Application Framework for R*, 2017. R package version 1.0.5.

[7] Jason Chuang and Ashley Jin. Termite — topic model visualization. `http://vis.stanford.edu/topic-diagnostics/model/silverStandards/`, 2012. Accessed on 2018-05-13.

[8] Jason Chuang, Christopher D Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the international working conference on advanced visual interfaces*, pages 74–77. ACM, 2012.

[9] Sana Malik, Alison Smith, Timothy Hawes, Panagis Papadatos, Jianyu Li, Cody Dunne, and Ben Shneiderman. Topicflow. `http://www.sanamalik.com/topicflow/TopicFlow.html#`, 2013. Accessed on 2018-05-01.

[10] Sana Malik, Alison Smith, Timothy Hawes, Panagis Papadatos, Jianyu Li, Cody Dunne, and Ben Shneiderman. Topicflow: visualizing topic alignment of twitter data over time. In *Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 720–726. ACM, 2013.

[11] Jaimie Murdock and Colin Allen. Visualization techniques for topic model checking. In *AAAI*, pages 4284–4285, 2015.

[12] Thomas Lin Pedersen. *shinyFiles: A Server-Side File System Viewer for Shiny*, 2016. R package version 0.6.2.

[13] Margaret E. Roberts, Brandon M. Stewart, and Dustin Tingley. *stm: R Package for Structural Topic Models*, 2017. R package version 1.3.0.

[14] Carson Sievert and Kenneth Shirley. Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70, 2014.

[15] Alison Smith, Timothy Hawes, and Meredith Myers. Hirarchie. `http://mlvl.github.io/Hierarchie/#/`, 2014. Accessed on 2018-04-26.

[16] Alison Smith, Timothy Hawes, and Meredith Myers. Hirarchie: Visualization for hierarchical topic models. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 71–78, 2014.

[17] Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392, 2005.

[18] Ramnath Vaidyanathan, Yihui Xie, JJ Allaire, Joe Cheng, and Kenton Russell. *htmlwidgets: HTML Widgets for R*, 2017. R package version 0.9.