Evaluating Visual Representations for Topic Understanding and Their Effects on Manually Generated Topic Labels

Alison Smith* **Tak Yeon Lee*** **Forough Poursabzi-Sangdeh†**
Jordan Boyd-Graber† **Niklas Elmqvist*** **Leah Findlater***
*University of Maryland, College Park, MD
†University of Colorado, Boulder, CO
{amsmit,tylee}@cs.umd.edu
{forough.poursabzisangdeh, jordan.boyd.graber}@colorado.edu
{elm,leahkf}@cs.umd.edu

Abstract

Probabilistic topic models are important tools for indexing, summarizing, and analyzing large document collections by their themes. However, promoting end-user understanding of topics remains an open research problem. We compare labels generated by users given four topic visualization techniques—word lists, word lists with bars, word clouds, and network graphs—against each other and against automatically generated labels. Our basis of comparison is participant ratings of how well labels describe documents from the topic. Our study has two phases: a labeling phase where participants label visualized topics and a validation phase where different participants select which labels best describe the topics' documents. Although all visualizations produce similar quality labels, simple visualizations such as word lists allow participants to quickly understand topics, while complex visualizations take longer but expose multi-word expressions that simpler visualizations obscure. Automatic labels lag behind user-created labels, but our dataset of manually labeled topics highlights linguistic patterns (e.g., hypernyms, phrases) that can be used to improve automatic topic labeling algorithms.

1 Comprehensible Topic Models Needed

A central challenge of the "big data" era is to help users make sense of large text collections (Hotho et al., 2005). A common approach to summarizing the main themes in a corpus is to use *topic models* (Blei, 2012), which are data-driven statistical models that identify words that appear together in similar documents. These sets of words or "topics" evince internal coherence and can help guide users to relevant documents. For instance, an FBI investigator sifting through the released Hillary Clinton e-mails may see a topic with the words "Benghazi", "Libya", "Blumenthal", and "success", spurring the investigator to dig deeper to find further evidence of inappropriate communication with longtime friend Sidney Blumenthal regarding Benghazi.

A key challenge for topic modeling, however, is how to promote end-user understanding of individual topics and the overall model. Most existing topic presentations use simple word lists (Chaney and Blei, 2012; Eisenstein et al., 2012). Although a variety of alternative topic visualization techniques exist (Sievert and Shirley, 2014; Yi et al., 2005), there has been no systematic assessment to compare them. Beyond exploring different visualization techniques, another means of making topics easier for users to understand is to provide descriptive labels to complement a topic's set of words (Aletras et al., 2014). Unfortunately, manual labeling is slow and, while automatic labeling approaches exist (Lau et al., 2010; Mei et al., 2007; Lau et al., 2011), their effectiveness is not guaranteed for all tasks.

To better understand these problems, we use *labeling* to evaluate topic model *visualizations*. Our study compares the impact of four commonly used topic visualization techniques on the labels that users create when interpreting a topic (Figure 1): word lists, word lists with bars, word clouds, and network graphs. On Amazon Mechanical Turk, one set of users viewed a series of individual topic vi-

Transactions of the Association for Computational Linguistics, vol. 5, pp. 1–16, 2017. Action Editor: Timothy Baldwin.
Submission batch: 2/2016; Revision batch: 6/2016; Published 1/2017.

sualizations and provided a label to describe each topic, while a second set of users assessed the quality of those labels alongside automatically generated ones.[1] Better labels imply that the topic visualization provide users a more accurate interpretation (labeling) of the topic.

The four visualization techniques have inherent trade-offs. Perhaps unsurprisingly, there is no meaningful difference in the quality of the labels produced from the four visualization techniques. However, simple visualizations (word list and word cloud) support a quick, first-glance understanding of topics, while more complex visualizations (network graph) take longer but reveal relationships between words. Also, user-created labels are better received than algorithmically-generated labels, but more detailed analysis uncovers features specific to high-quality labels (e.g., tendency towards abstraction, inclusion of phrases) and the types of topics for which automatic labeling works. These findings motivate future automatic labeling algorithms.

2 Background

Presenting the full text of a document corpus is often impractical. For truly large and complex text corpora, abstractions, such as topic models, are necessary. Here we review probabilistic topic modeling and topic model interfaces.

2.1 Probabilistic Topic Modeling

Topic modeling algorithms produce statistical models that discover key themes in documents (Blei, 2012). Many specific algorithms exist; in this work we use Latent Dirichlet Allocation (Blei et al., 2003, LDA) as it is commonly employed. LDA is an unsupervised statistical topic modeling algorithm that considers each document to be a "bag of words" and can scale to large corpora (Zhai et al., 2012; Hoffman et al., 2013; Smola and Narayanamurthy, 2010). Assuming that each document is an admixture of topics, inference discovers each topic's distribution over words and each document's distribution over topics that best explain the corpus. The set of topics provide a high-level overview of the corpus, and individual topics can link back to the original documents to support directed exploration. The topic distributions can also be used to present other documents related to a given document.

Clustering is hard because there are multiple reasonable objectives that are impossible to satisfy simultaneously (Kleinberg, 2003). Topic modeling evaluation has focused on *perplexity*, which measures how well a model can predict words in unseen documents (Wallach et al., 2009b; Jelinek et al., 1977). However, Chang et al. (2009) argue that evaluations optimizing for perplexity encourage complexity at the cost of human interpretability. Newman et al. (2010a) build on this insight, noting that "one indicator of usefulness is the ease by which one could think of a short label to describe the topic." Unlike previous interpretability studies, here we examine the connection between a topic's *visual representation* (not just its content) and its interpretability.

Recent work has focused on automatic generation of labels for topics. Lau et al. (2011) use Wikipedia articles to automatically label topics. The assumption is that for each topic there will be a Wikipedia article title that offers a good representation of the topic. Aletras et al. (2014) use a graph-based approach to better rank candidate labels. They generate a graph from the words in candidate articles and use PageRank to find a representative label. In Section 3 we use an adapted version of the method presented by Lau et. al. (2011) as a representative automatic labeling algorithm.

2.2 Topic Model Visualizations

The topic visualization techniques in our study—word list, word list with bars, word cloud, and network graph—commonly appear in topic modeling tools. Here, we provide an overview of tools that display an *entire topic model or models* to the user, while more detail on the *individual topic* visualization techniques can be found in Section 3.2.

Topical Guide (Gardner et al., 2010), Topic Viz (Eisenstein et al., 2012), and the Topic Model Visualization Engine (Chaney and Blei, 2012) are tools that support corpus understanding and directed browsing through topic models. They display the model overview as an aggregate of underlying topic visualizations. For example, Topical Guide uses hor-

[1] Data available at `https://github.com/alisonmsmith/Papers/tree/master/TopicRepresentations`.

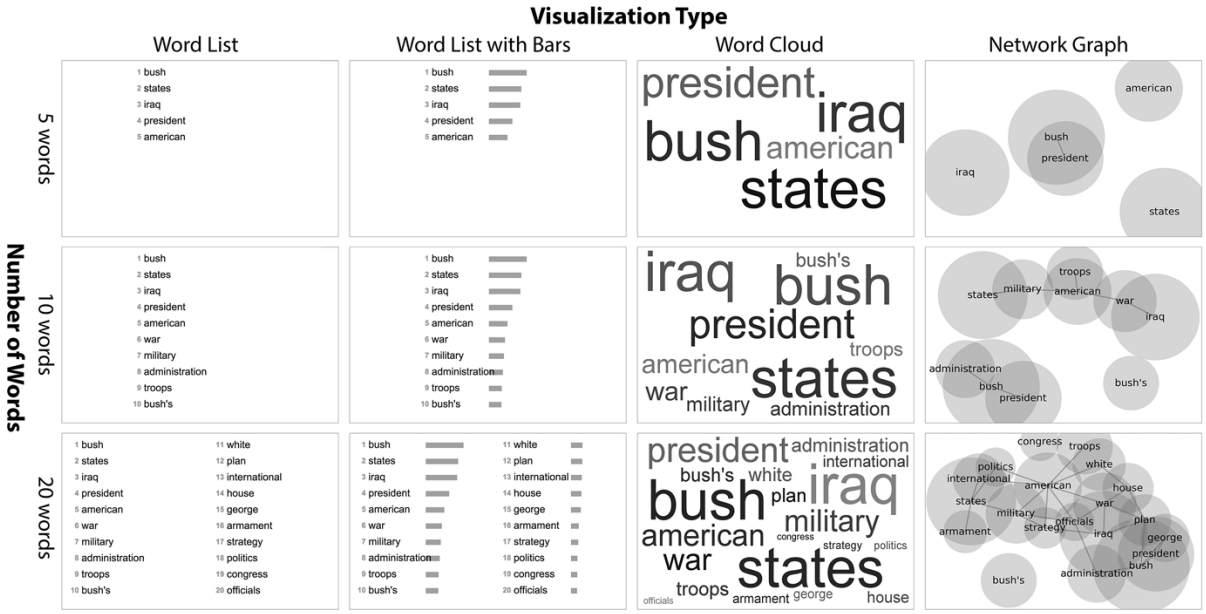


Figure 1: Examples of the twelve experimental conditions, each a different visualization of the same topic about the George W. Bush presidential administration and the Iraq War. Rows represent cardinality, or number of topic words shown (five, ten, twenty). Columns represent visualization techniques. For **word list** and **word list with bars**, topic words are ordered by their probability for the topic. **Word list with bars** also includes horizontal bars to represent topic-term probabilities. In the **word cloud**, words are randomly placed but are sized according to topic-term probabilities. The **network graph** uses a force-directed layout algorithm to co-locate words that frequently appear together in the corpus.

izontal **word lists** when displaying an overview of an entire topic model but uses a **word cloud** of the top 100 words for a topic when displaying only a single topic. Topic Viz and the Topic Model Visualization Engine both represent topics with vertical **word lists**; the latter also uses set notation.

Other tools provide additional information within topic model overviews, such as the relationship between topics or temporal changes in the model. However, they still require the user to understand individual topics. LDAVis (Sievert and Shirley, 2014) includes information about the relationship between topics in the model. Multi-dimensional scaling projects the model's topics as circles onto a two-dimensional plane based on their inter-topic distances; the circles are sized by their overall prevalence. The individual topics, however, are then visualized on demand using a **word list with bars**. Smith et al. (2014) visualize a topic model using a nested network graph layout called group-in-a-box (Rodrigues et al., 2011, GIB). The individual topics are displayed using a **network graph** visualization, and related topics are displayed within a treemap (Shneiderman, 1992) layout. The result is a visualization where related words cluster within topics and related topics cluster in the overall layout.

TopicFlow (Smith et al., 2015) visualizes how a model changes over time using a Sankey diagram (Riehmann et al., 2005). The individual topics are represented both as **word lists** in the model overview and as **word list with bars** when viewing a single topic or comparing between two topics. Argviz (Nguyen et al., 2013) captures temporal shifts in topics during a debate or a conversation. The individual topics are presented as **word lists** in the model overview and using **word list with bars** for the selected topics. Klein et al. (2015) use a *dust-and-magnet* visualization (Yi et al., 2005) to visualize the force of topics on newspaper issues. The temporal trajectories of several newspapers are displayed as dust trails in the visualization. The individual topics are displayed as **word clouds**.

In contrast to these visualizations which support viewing the underlying topics on demand, Termite (Chuang et al., 2012) uses a tabular layout of words and topics to provide an overview of the model to compare *across* topics. It organizes the model into clusters of *related* topics based on word overlap. This clustered representation is both space-efficient and speeds corpus understanding.

Despite the breadth of topic model visualizations, a small set of individual topic representations are ubiquitous: word list, word list with bars, word cloud, and network graph. In the following sections, we compare these topic visualization techniques.

3 Method: Comparing Visualizations

We conduct a controlled online study to compare the four commonly used visualization techniques identified in Section 2: word list, word list with bars, word cloud, and network graph. We also compare effectiveness with the number of topic words shown, that is, the *cardinality* of the visualization: five, ten or twenty topic words.

3.1 Dataset

We select a corpus that does not assume domain expertise: 7,156 *New York Times* articles from January 2007 (Sandhaus, 2008). We model the corpus using an LDA (Blei et al., 2003) implementation in Mallet (Yao et al., 2009) with domain-specific stopwords and standard hyperparameter settings.[2] Our simple setup is by design: our goal is to emulate the "off the shelf" behavior of conventional topic modeling tools used by novice users. Instead of improving the quality of the model using asymmetric priors (Wallach et al., 2009a) or bigrams (Boyd-Graber et al., 2014), our topic model has topics of variable quality, allowing us to explore the relationship between topic quality and our task measures.

Automatic labels are generated from representative Wikipedia article titles using a technique similar to Lau et al. (2011). We first index Wikipedia using Apache Lucene.[3] To label a topic, we query Wikipedia with the top twenty topic words to retrieve fifty articles. These articles' titles comprise our candidate set of labels. We then represent each article using its TF-IDF vector and calculate the centroid (average TF-IDF) of the retrieved articles. To rank and choose the most representative of the set, we calculate the cosine similarity between the centroid TF-IDF vector and the TF-IDF vector of each of the articles. We choose the title of the article with the maximum cosine similarity to the centroid. Unlike Lau et al. (2011), we do not include the topic words or Wikipedia title n-grams derived from our label set, as these labels are typically not the best candidates. Although other automatic labeling techniques exist, we choose this one as it is representative of general techniques.

3.2 Visualizations

As discussed in Section 2, our study compares four of the most common topic visualization techniques. To produce a meaningful comparison, the space given to each visualization is held constant: 400 $\times$ 250 pixels. Figure 1 shows each visualization for the three cardinalities (or number of words displayed) for the same topic.

Word List The most straightforward topic representation is a list of the top n words in the topic, ranked by their probability. In practice, topic word lists have many variations. They can be represented horizontally (Gardner et al., 2010; Smith et al., 2015) or vertically (Eisenstein et al., 2012; Chaney and Blei, 2012), with or without commas separating the individual words, or using set notation (Chaney and Blei, 2012). Nguyen et al. (2013) add the weights to the word list by sizing the words based on their probability for the topic, which blurs the boundary with word clouds; however, this approach is not common. We use a horizontal list of equally sized words ordered by the probability $p(w|z)$ for the word w in the topic z. For space efficiency, we organize our word list in two columns and add item numbers to make the ordering explicit.

Word List with Bars Combining bar graphs with word lists yields a visual representation that not only conveys the ordering but also the absolute value of the weights associated with the words. We use a similar implementation to Smith et al. (2015) to add horizontal bars to the word list for a topic z where the length of each bar represents the probability $p(w|z)$ for each word w.

[2] n=50, α=0.1, β=0.01

[3] `http://lucene.apache.org/`

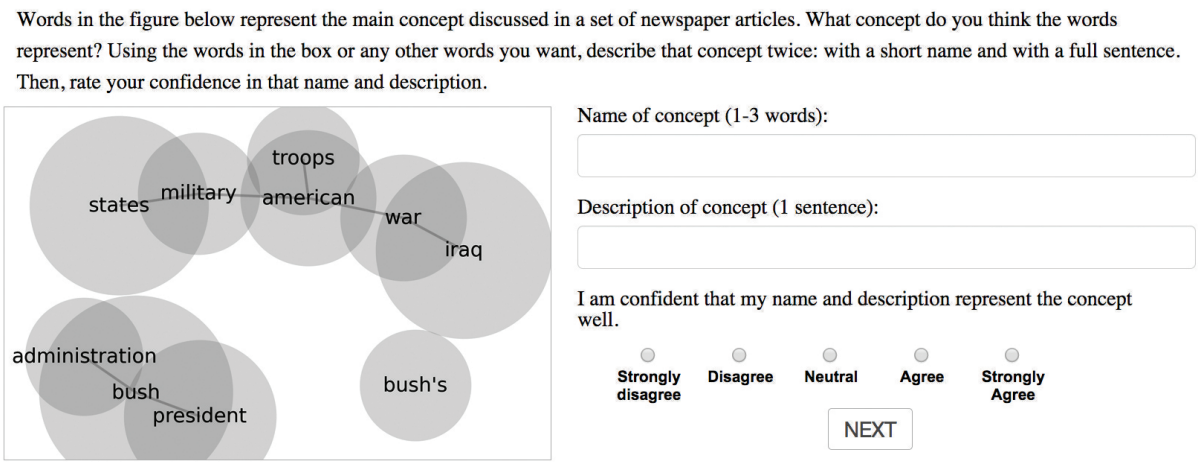


Figure 2: The labeling task for the network graph and ten words. Users create a short label and full sentence describing the topic and rate their confidence that the label and sentence represent the topic well.

Word Cloud The word cloud (or *tag cloud*) is one of the most popular and well-known text visualization techniques and is a common visualization for topics. Many options exist for word cloud layout, color scheme, and font size (Mueller, 2012). Existing work on layouts is split between those that size words by their frequency or probability for the topic (Ramage et al., 2010) and those that size by the rank order of the word (Barth et al., 2014). We use a combination of these techniques where the word's font size is initially set proportional to its probability in a topic $p(w|z)$. However, when the word is too large to fit in the canvas, the size is gradually decreased (Barth et al., 2014). We use a gray scale to visually distinguish words and display all words horizontally to improve readability.

Network Graph Our most complex topic visualization is a network graph. We use a similar network graph implementation to Smith et al. (2014), which represents each topic as a node-link diagram, where words are circular nodes with edges drawn between commonly co-occurring words. Each word's radius is scaled by the probability $p(w|z)$ for the word w in a topic z. While Smith et al. (2014) draw edges based on document-level co-occurrence, we instead use edges to pull together phrases, so they are drawn between words w_1 and w_2 based on bigram count, specifically if $\log(count(w_1,w_2)) > k$, with $k = 0.1$.[4] Edge width and color are applied uniformly to further reduce complexity in the graph. The network graph is displayed using a force-directed graph layout algorithm (Fruchterman and Reingold, 1991) where all nodes repel each other but links attract connected nodes.

3.3 Cardinality

Although every word has some probability for every topic, $p(w|z)$, visualizations typically display only the top n words. The cardinality may interact with the effectiveness of the different visualization techniques (e.g., more complicated visualizations may degrade with more words). We use $n \in \{5, 10, 20\}$.

3.4 Task and Procedure

The study includes two phases with different users. In **Labeling** (Phase I), users describe a topic given a specific visualization, and we measure speed and self-reported confidence in completing the task. In **Validation** (Phase II), users select the best and worst among a set of Phase I descriptions and an automatically generated description for how well they represent the original topics' documents.

Phase I: Labeling For each labeling task, users see a topic visualization, provide a short *label* (up

[4] From $k \in \{0.01, 0.05, 0.1, 0.5\}$, we chose $k = 0.1$ as the best trade-off between complexity and provided information.

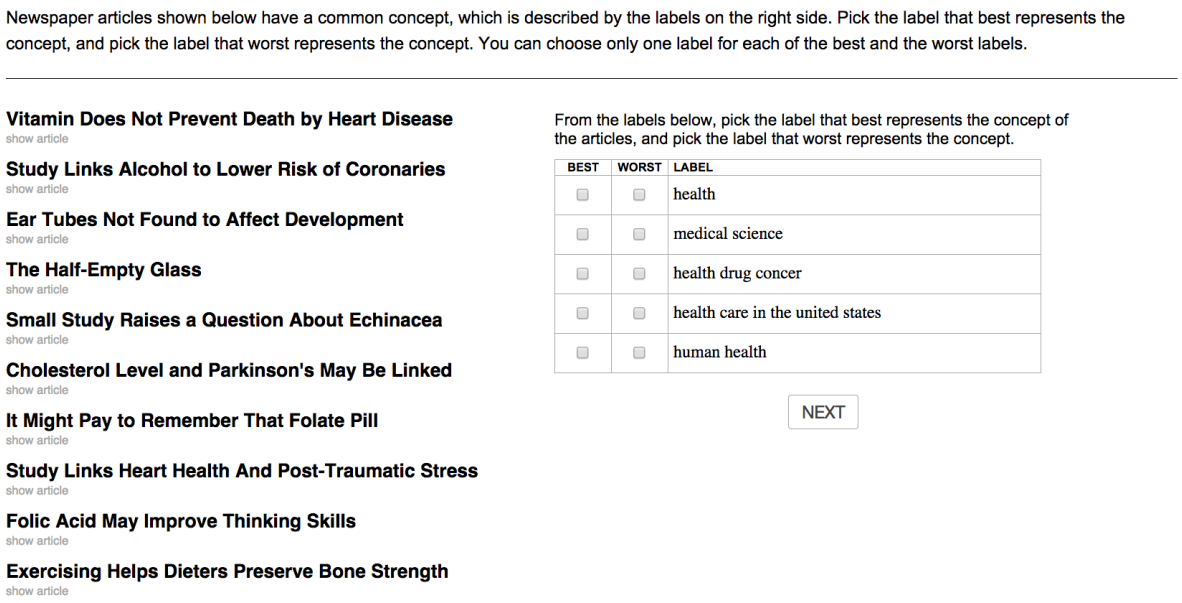


Figure 3: The validation task shows the titles of the top ten documents and five potential labels for a topic. Users are asked to pick the best and worst labels. Four labels were created by Phase I users after viewing different visualizations of the topic, while the fifth was generated by the algorithm. The labels are shown in random order.

to three words), then give a longer *sentence* to describe the topic, and finally use a five-point Likert scale to rate their confidence that the label and sentence represent the topic well. We also track the time to perform the task. Figure 2 shows an example of a labeling task using the network graph visualization technique with ten words.

Labeling tasks are randomly grouped into human intelligence tasks (HIT) on Mechanical Turk[5] such that each HIT includes five tasks from the same visualization technique.[6]

Phase II: Validation In the validation phase, a new set of users assesses the quality of the labels and sentences created in Phase I by evaluating them against documents associated with the given topic. It is important to evaluate the topic labels in *context*; a label that superficially looks good is useless if it is not representative of the underlying documents in the corpus. Algorithmically generated labels (not sentences) are also included. Figure 3 shows an example of the validation task.

The user-generated labels and sentences are evaluated separately. For each task, the user sees the titles of the top ten documents associated with a topic and a randomized set of labels or sentences, one elicited from each of the four visualization techniques within a given cardinality. The set of labels also includes an algorithmically generated label. We ask the user to select the "best" and "worst" of the labels or sentences based on how well they describe the documents. Documents are associated to topics based on the probability of the topic, z, given the document, d, $p(z|d)$. Only the title of each document is initially shown to the user with an option to "show article" (or view the first 400 characters of the document).

All labels are lowercased to enforce uniformity. We merge identical labels so users do not see duplicates. If a merged label receives a "best" or "worst" vote, the vote is split equally across all of the original instances (i.e., across multiple visualization techniques with that label). Finally, we track task com-

[5] All users are in the US or Canada, have more than fifty previously approved HITs, and have an approval rating greater than 90%.

[6] We did not restrict users from performing multiple HITs, which may have exposed them to multiple visualization techniques. Users completed on average 1.5 HITs.

pletion time.

Each user completes four randomly selected validation tasks as part of a HIT, with the constraint that each task must be from a different topic. We also use ground truth seeding for quality control: each HIT includes one additional test task that has a purposefully bad label generated by concatenating three random dictionary words. If the user does not pick the bad label as the "worst", we discard all data in that HIT.

3.5 Study Design and Data Collection

For Phase I, we use a factorial design with factors of *Visualization* (levels: word list, word list with bars, word cloud, and network graph) and *Cardinality* (levels: 5, 10, and 20), yielding twelve conditions. For each of the fifty topics in the model and each of the twelve conditions, at least five users perform the labeling task, describing the topic with a label and sentence, resulting in a minimum of 3,000 label and sentence pairs. Each HIT includes five of these labeling tasks, for a minimum of 600 HITs. The users are paid \$0.30 per HIT.

For Phase II, we compare descriptions across the four visualization techniques (and automatically generated labels), but only *within* a given cardinality level rather than *across* cardinalities. We collected 3,212 label and sentence pairs from 589 users during Phase I. For validation in Phase II, we use the first five labels and sentences collected for each condition for a total of 3.000 labels and sentences. These are shown in sets of four (labels or sentences) during Phase II, yielding a total of 1,500 ($3{,}000/4$ + $3{,}000/4$) tasks. Each HIT contains four validation tasks and one ground truth seeding task, for a total of 375 HITs. To increase robustness, we validate twice for a total of 750 HITs, without allowing any two labels or sentences to be compared twice. The users get \$0.50 per HIT.

4 Results

We analyze labeling time and self-reported confidence for the labeling task (Phase I) before reporting on the label quality assessments (Phase II). We then analyze linguistic qualities of the labels, which should motivate future work in automatic label generation.

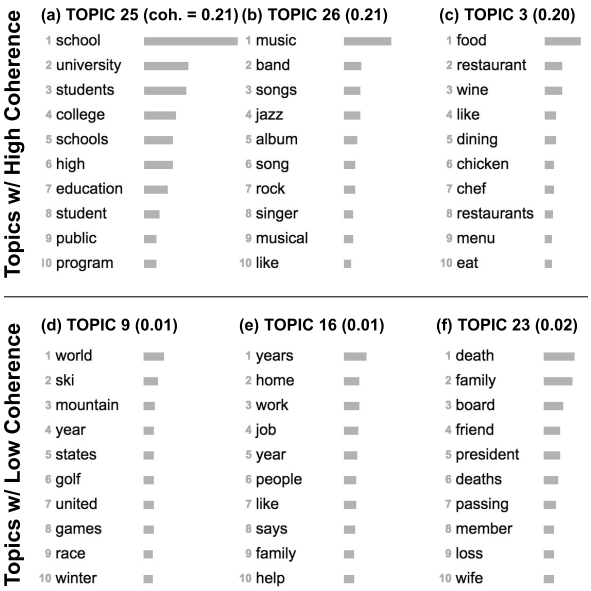


Figure 4: Word list with bar visualizations of the three best (top) and worst (bottom) topics according to their coherence score, which is shown to the right of the topic number. The average topic coherence is 0.09 (SD=0.05).

We first provide an example of user-generated labels and sentences: the user labels for the topic shown in Figure 1 include government, iraq war, politics, bush administration, and war on terror. Examples of sentences include "President Bush's military plan in Iraq" and "World news involving the US president and Iraq".[7]

To interpret the results, it is useful to also understand the quality of the generated topics, which varies throughout the model and may impact a user's ability to generate good labels. We measure topic quality using *topic coherence*, an automatic measure that correlates with how much sense a topic makes to a user (Lau et al., 2014).[8] The average topic coherence for the model is 0.09 ($SD = 0.05$). Figure 4 shows the three best (top) and three worst topics (bottom) according to their observed coherence: the coherence metric distinguishes obvious topics from inscrutable ones. Section 4.3 shows that users cre-

[7]The complete set of labels and sentences are available at `https://github.com/alisonmsmith/Papers/tree/master/TopicRepresentations`.

[8]We use a reference corpus of 23 million Wikipedia articles for computing normalized pointwise mutual information needed for computing the observed coherence.

Technique	Word List			Word List w/ Bars			Word Cloud			Network Graph		
Cardinality	5	10	20	5	10	20	5	10	20	5	10	20
# tasks completed	264	268	268	264	280	260	268	268	268	267	274	263
Avg time (*SD*)	53.0 (44.3)	53.2 (46.6)	52.1 (53.3)	58.4 (75.1)	58.7 (51.1)	60.7 (57.9)	52.7 (47.4)	49.4 (37.4)	68.4 (85.4)	55.0 (50.7)	55.6 (56.0)	77.9 (71.9)
Avg confidence (*SD*)	3.7 (0.9)	3.7 (0.9)	3.6 (0.9)	3.6 (0.9)	3.6 (0.8)	3.7 (0.8)	3.5 (1.0)	3.6 (0.9)	3.6 (0.9)	3.4 (1.1)	3.6 (0.8)	3.7 (0.8)

Table 1: Overview of the labeling phase: number of tasks completed, the average and standard deviation (in parentheses) for time spent per task in seconds, and the average and standard deviation for self-reported confidence on a 5-point Likert scale for each of the twelve conditions.

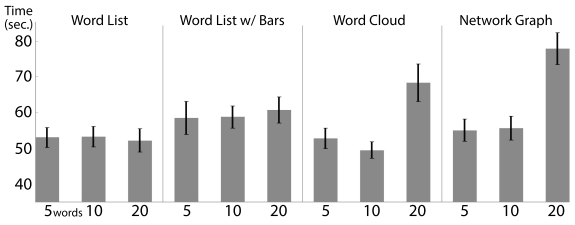


Figure 5: Average time for the labeling task, across visualizations and cardinalities, ordered from left to right by visual complexity. For 20 words, network graph was significantly slower and word list was significantly faster than the other visualization techniques. Error bars show standard error.

ated lower quality labels for low coherence topics.

4.1 Labeling Time

More complex visualization techniques take longer to label (Table 1 and Figure 5). The labeling tasks took on average 57.9 seconds ($SD = 58.5$) to complete and a two-way ANOVA (visualization technique $\times$ cardinality) reveals significant main effects for both the visualization technique[9] and the cardinality,[10] as well as a significant interaction effect.[11]

For lower cardinality, the labeling time across visualization techniques is similar, but there are notable differences for higher cardinality. Posthoc pairwise comparisons based on the interaction effect (with Bonferroni adjustment) found no significant differences between visualizations with five words and only one significant difference for ten words (word list with bars was slower than word cloud, $p < .05$). For twenty words, however, the network graph was significantly slower at an average of 77.9s ($SD = 72.0$) than the other three visualizations ($p < .05$). This effect is likely due to the network graph becoming increasingly dense with more nodes (Figure 1, *bottom right*). In contrast, the relatively simple word list visualization was significantly faster with twenty words than the three other visualizations ($p < .05$), taking only 52.1s on average ($SD = 53.4$). Word list with bars and word cloud were not significantly different from each other.

As a secondary analysis, we examine the relationship between elapsed time and the observed coherence for each topic. Topics with high coherence scores, for example, may be faster to label, because they are easier to interpret. However, the small negative correlation between time and coherence (Figure 6, *top*) was not significant ($r_{48} = -.13$, $p = .364$).

4.2 Self-Reported Labeling Confidence

For each labeling task, users rate their confidence that their labels and sentences describe the topic well on a scale from 1 (least confident) to 5 (most confident). The average confidence across all conditions was 3.6 ($SD = 0.9$). Kruskal-Wallis tests show a significant impact of visualization technique on confidence with five and ten words, but not twenty.[12] While average confidence ratings across all conditions only range from 3.4 to 3.7, perceived confidence with network graph suffers when the visualization has too few words (Table 1).

As a secondary analysis, we compare the self-reported confidence with observed coherence for each topic (Figure 6, *bottom*). Increased user confidence with more coherent topics is supported by a moderate positive correlation between topic coher-

[9] $F_{(3,3199)} = 10.58$, $p < .001$, $\eta_p^2 = .01$
[10] $F_{(2,3199)} = 14.60$, $p < .001$, $\eta_p^2 = .01$
[11] $F_{(6,3199)} = 4.59$, $p < .001$, $\eta_p^2 = .01$

[12] Five words: $\chi_3^2 = 12.62$, $p = .006$. Ten words: $\chi_3^2 = 7.94$, $p = .047$. We used nonparametric tests because the data is ordinal and we cannot guarantee that all differences between points on the scale are equal.

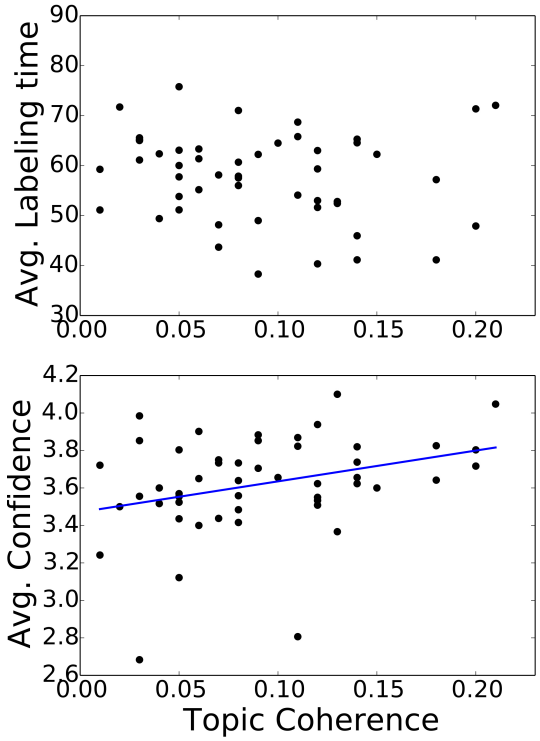


Figure 6: Relationship between observed coherence and labeling time (top) and observed coherence and self-reported confidence (bottom) for each topic. The positive correlation (Slope = 1.64 and R^2 = 0.10) for confidence is significant.

ence and confidence ($r_{48} = .32$, $p = .026$). This result provides further evidence that topic coherence is an effective measurement of topic interpretability.

4.3 Other Users' Rating of Label Quality

Other users' perceived quality of topic labels is the best real-world measure of quality (as described in Section 3.4). Overall, the visualization techniques had similar quality labels, but automatically generated labels do not fare well. Automatic labels get far fewer "best" votes and far more "worst" votes than user-generated labels produced from any of the four visualization techniques (Figure 7). Chi-square tests on the distribution of "best" votes for labels for each cardinality show that the visualization matters.[13] Posthoc analysis using pairwise Chi-square tests with Bonferroni correction show that automatic labels were significantly worse than user-generated labels from each of the visualization techniques (all comparisons $p < .05$). No other pairwise comparisons were significant.

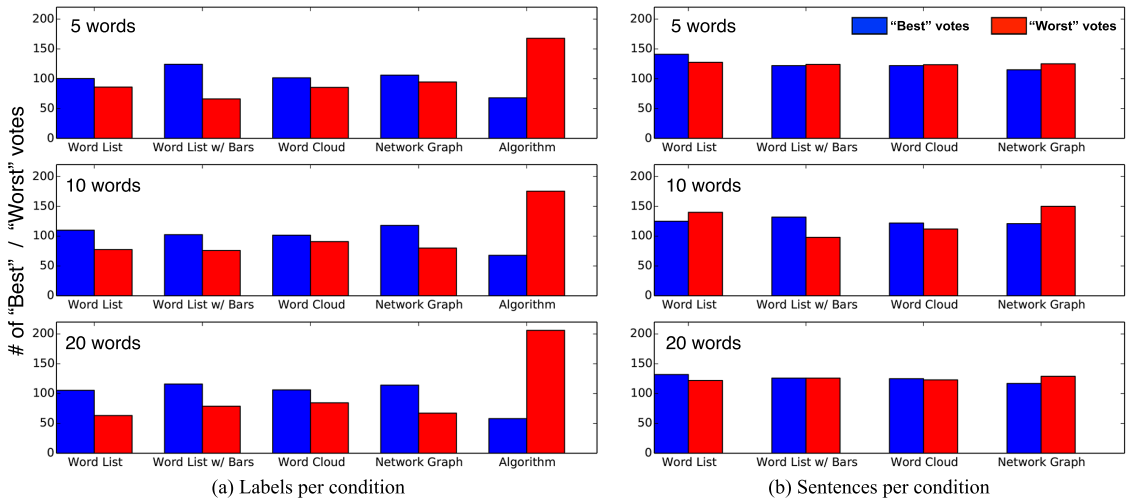
For sentences, no visualization technique emerged as better than the others. Additionally, there is no existing automatic approach to compare against. The distribution of "best" counts here was relatively uniform. Separate Kruskal-Wallis tests for each cardinality to examine the impact of the visualization techniques on "best" counts did not reveal any significant results.

As a secondary qualitative analysis, we examine the relationship between topic coherence and the assessed quality of the labels. The automatic algorithm tended to produce better labels for the coherent topics than for the incoherent topics. For example, Topic 26 (Figure 4, b)—{music, band, songs}—and Topic 31 (Figure 4, c)—{food, restaurant, wine}—are two of the most coherent topics. The automatic algorithm labeled Topic 26 as music and Topic 31 as food. For both of these coherent topics, the labels generated by the automatic algorithm secured the most "best" votes and no "worst" votes. In contrast, Topic 16 (Figure 4, e)—{years, home, work}—and Topic 23 (Figure 4, f)—{death, family, board}—are two of the least coherent topics. The automatic labels refusal of work and death of michael jackson yielded the most "worst" votes and fewest "best" votes.

To further demonstrate this relationship, we extracted from the 50 topics the top and bottom *quartiles* of 13 topics each[14] based on their observed coherence scores. Figure 8 shows a comparison of the "best" and "worst" votes for the topic labels for these quartiles, including user-generated and automatically generated labels. For the top quartile, the number of "best" votes per technique ranged from 61 for automatic labels to 96 for the network graph visualization. The range for the bottom quartile was larger, from only 45 "best" votes for automatic labels to 99 for word list with bars. The automatic labels, in particular, received a large relative increase in "best" votes when comparing the bottom quartile

[13] Five words: $\chi^2_{4,N=500} = 16.47$, $p = .002$. Ten words: $\chi^2_{4,N=500} = 14.62$, $p = .006$. Twenty words: $\chi^2_{4,N=500} = 22.83$, $p < .001$.

[14] We could not get exact quartiles, because we have 50 topics, so we rounded up to include 13 topics in each quartile.



(a) Labels per condition (b) Sentences per condition

Figure 7: The "best" and "worst" votes for labels and sentences for each condition. The automatically generated labels received more "worst" votes and fewer "best" votes compared to the user-created labels.

to the top quartile (increase of 37%).

Additionally, the word list, word cloud, and network graph visualizations all lead to labels with similar "best" and "worst" votes for both the top and bottom quartiles. However, the word list with bars representation shows both a large relative increase for the best votes (increase of 19%) and relative decrease for the "worst" votes (decrease of 23%) when comparing the top to the bottom quartile. These results suggest that adding numeric word probability information highlighted by the bars may help users understand poor quality topics.

4.4 Label Analysis

The results of Phase I provide a large manually generated label set. Exploratory analysis of these labels reveals linguistic features users tend to incorporate when labeling topics. We discuss implications for automatic labeling in Section 5. In particular, users prefer shorter labels, labels that include topic words and phrases, and abstraction in topic labeling.

Length The manually generated labels use 2.01 words ($SD = 0.95$), and the algorithmically generated labels use 3.16 words ($SD = 2.05$). Interestingly, the labels voted as "best" were shorter on average than those voted "worst", regardless of whether algorithmically generated labels are included in the analysis. With algorithmically generated labels included, the average lengths are 2.04 ($SD = 1.16$) words for "best" labels and 2.83 ($SD = 1.79$) words for "worst" labels,[15] but even without the algorithmically generated labels, the "best" labels are

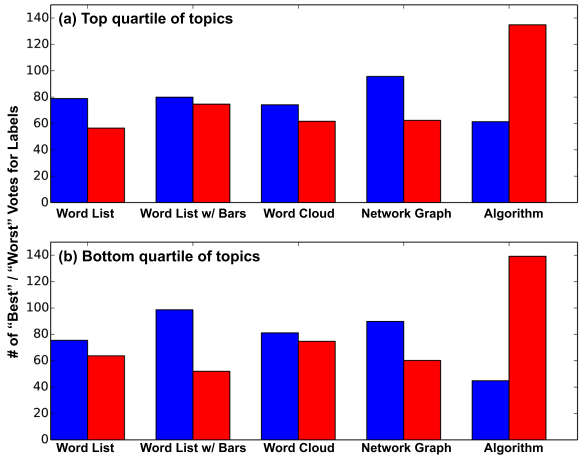


Figure 8: Comparison of the "best" and "worst" votes for labels generated using the different visualization techniques (and the automatically generated labels) for the top quartile of topics (top) and bottom quartile of topics (bottom) by topic coherence. The automatically generated labels receive far more "best" votes for the coherent topics.

[15]The "best" label set includes all labels voted at least once as "best", and similarly the "worst" label set includes all labels voted at least once as "worst".

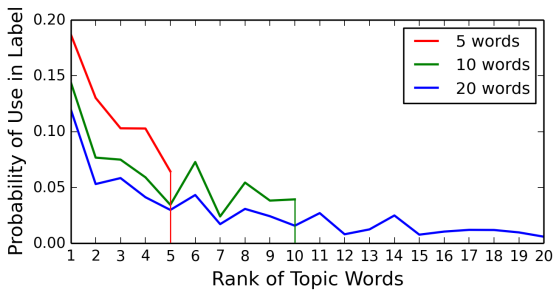


Figure 9: Relationship between rank of topic words and the average probability of occurrences in labels. The three lines—red, green, and blue—represent cardinality of five, ten, and twenty, respectively. The higher-ranked words were used more frequently.

shorter ($M = 1.96$, $SD = .87$) than the "worst" labels ($M = 2.09$, $SD = 1.01$).

Shared Topic Words Of the 3,212 labels, 2,278, or 71%, contain at least one word taken directly from the topic words—that is, the five, ten, or twenty words shown in the visualization; however, there are no notable differences between the visualization techniques. Additionally, the number of topic words included on average was similar across all three cardinalities, suggesting that users often use the same number of topic words regardless of how many were shown in the visualization.

We further examine the relationship between a topic word's rank and whether the word was selected for inclusion in the labels. Figure 9 shows the average probability of a topic word being used in a label by the topic word's rank. More highly ranked words were included more frequently in labels. As cardinality increased, the highest ranked words were also less likely to be employed, as users had more words available to them.

Phrases Although LDA makes a "bag of words" assumption when generating topics, users can reconstruct relevant phrases from the unique words. For Topic 26, for example, all visualizations include the same topic terms. However, the network graph visualization highlights the phrases "jazz singer" and "rock band" by linking their words as commonly co-occurring terms in the corpus. These phrases are not as easily discernible in the word cloud visualization (Figure 10). We compute a set of common

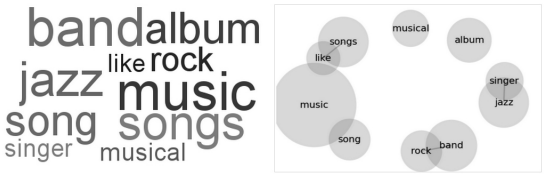


Figure 10: Word cloud and network graph visualizations of Topic 26. Phrases such as "jazz singer" and "rock band" are obscured in the word cloud but are shown in the network graph as connected nodes.

phrases by taking all bigrams and trigrams that occur more than fifty and twenty times, respectively, in the NYT corpus. Of the 3212 labels, 575 contain one of these common phrases, but those generated by users with the network graph visualization contain the most phrases. Labels generated in the word list (22% of the labels), word list with bars (25%), and word cloud (24%) conditions contain fewer phrases than the labels generated in the network graph condition (29%). Although it is not surprising that the network graph visualization better communicates common phrases in the corpus as edges are drawn between these phrases, this suggests other approaches to drawing edges. Edges drawn based on sentence or document-based co-occurrence, for example, could instead uncover longer-distance dependencies between words, potentially identifying distinct subtopics with a topic.

Hyponymy Users often prefer more general terms for labels than the words in the topic (Newman et al., 2010b). To measure this, we look for the set of unique hyponyms and hypernyms of the topic words, or those that are not themselves a topic word, that appear in the manually generated labels. We use the super-subordinate relation, which represents hypernymy and hyponymy, from WordNet (Miller, 1995). Of the 3,212 labels, 235 include a unique hypernym and 152 include a unique hyponym of the associated topic words found using WordNet, confirming that users are significantly more likely to produce a more generic description of the topic ($\chi^2_{1,N=387} = 17.38$, $p < .001$). For the 235 more generic labels, fewer of these came from word list (22%) and more from the network graph (30%) than the other visualization techniques—word list with bars (24%) and word cloud (24%). This may mean

that the network graph helps users to better understand the topic words as a group and therefore label them using a hypernym. We also compared hypernym inclusion for "best" and "worst" labels: 63 (5%) of the "best" labels included a hypernym while only 44 (3%) of the "worst" labels included a hypernym. Each of the visualization techniques led to approximately the same percentage of the 152 total more specific labels.

5 Discussion

Although the four visualization techniques yield similar quality labels, our crowdsourced study highlights the strengths and weaknesses of the techniques. It also reveals some preferred linguistic features of user-generated labels and how these differ from automatically generated labels.

The trade-offs among the visualization techniques show that context matters. If efficiency is paramount, then word lists—both simple and fast—are likely best. For a cardinality of twenty words, for example, users presented with the simple word list are significantly faster at labeling than those shown the network graph visualization. At the same time, more complex visualizations expose users to multi-word expressions that the simpler visualization techniques may obscure (Section 4.4). Future work should investigate for what types of user tasks this information is most useful. There is also potential for misinterpretation of topic meaning when cardinality is low. Users can misunderstand the topic based on the small set of words, or adjacent words can inadvertently appear to form a meaningful phrase, which may be particularly an issue for the word cloud.

Our crowdsourced study identified the "best" and "worst" labels for the topic's documents. An additional qualitative coding phase could evaluate each "worst" label to determine why, whether due to misinterpretation, spelling or grammatical errors, length, or something else.

Surprisingly, we found no relationship between topic coherence and labeling time (Section 4.1). This is perhaps because not only are users quick to label topics they understand, but they also quickly give up when they have no idea what a topic is about. We do, however, find a relationship between coherence and confidence (Section 4.2). This positive correlation supports topic coherence as an effective measure for human interpretability.

Automatically generated labels are consistently chosen as the "worst" labels, although they are competitive with the user-generated labels for highly coherent topics (Section 4.3). Future automatic labeling algorithms should still be robust to poor topics. Algorithmically generated labels were longer and more specific than the user-generated labels. It is unsurprising that these automatic labels were consistently deemed the worst. Users prefer shorter labels with more general words (e.g., hypernyms, Section 4.4). We show specific examples of this phenomenon from Topic 14 and Topic 48. For Topic 14—{health, drug, medical, research, conditions}—the algorithm generated the label health care in the united states, but users preferred the less specific labels health and medical research. Similarly, for Topic 48—{league, team, baseball, players, contract}—the algorithm generated the label major league baseball on fox; users preferred simpler labels, such as baseball. Automatic labeling algorithms thus can be improved to focus on general, shorter labels. Interestingly, simple textual labels have been shown to be more efficient but less effective than topic keywords (i.e., word lists) for an automatic document retrieval task (Aletras and Stevenson, 2014), highlighting the extra information present in the word lists. Our findings show that users are also able to effectively interpret the word list information, as that visualization was both efficient and effective for the task of topic labeling compared to the other more complex visualizations.

Although we use WordNet to verify that users prefer more general labels, this is not a panacea, because WordNet does not capture all of the generalization users want in labels. In many cases, users use terms that synthesize relationships beyond trivial WordNet relationships, such as locations or entities. For example, Topic 18—{san, los, angeles, terms, francisco}—was consistently labeled as the location California, and Topic 38—{open, second, final, won, williams}—which almost all users labeled as tennis, required a knowledge of the entities Serena Williams and the U.S. Open. In addition to WordNet, an automatic labeling algorithm could

use a gazetteer for determining locations from topic words and a knowledge base such as TAP (Guha and McCool, 2003), which provides a broad range of information about popular culture for matching topic words to entities.

6 Conclusion and Future Work

We present a crowdsourced user study to compare four topic visualization techniques—a simple ranked word list, a ranked word list with bars representing word probability, a word cloud, and a network graph—based on how they impact the user's understanding of a topic. The four visualization techniques lead to similar quality labels as rated by end users. However, users label more quickly with the simple word list, yet tend to incorporate phrases and more generic terminology when using the more complex network graph. Additionally, users feel more confident labeling coherent topics, and manual labels far outperform the automatically generated labels against which they were evaluated.

Automatic labeling can benefit from this research in two ways: by suggesting when to apply automatic labeling and by providing training data for improving automatic labeling. While automatic labels falter compared to human labels in general, they do quite well when the underlying topics are of high quality. Thus, one reasonable strategy would be to use automatic labels for a portion of topics, but to use human validation to either first improve the remainder of the topics (Hu et al., 2014) or to provide labels (as in this study) for lower quality topics. Moreover, our labels provide training data that may be useful for automatic labeling techniques using feature-based models (Charniak, 2000)—combining information from Wikipedia, WordNet, syntax, and the underlying topics—to reproduce the types of labels and sentences created (and favored) by users.

Finally, our study focuses on comparing individual topic visualization techniques. An open question that we do not address is whether this generalizes to understanding entire topic models. In other words, simple word list visualizations are useful for quick and high-quality topic summarization, but does this mean that a collection of word lists—one per topic—will also be optimal when displaying the entire model? Future work should look at comparing visualization techniques for full topic model understanding.

Acknowledgments

We would like to thank the anonymous reviewers as well as the TACL editors, Timothy Baldwin and Lillian Lee, for helpful comments on an earlier draft of this paper. This work was funded by NSF grant IIS-1409287. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Nikolaos Aletras and Mark Stevenson. 2014. Labelling topics using unsupervised graph-based methods. In *Proceedings of the Association for Computational Linguistics*.

Nikolaos Aletras, Timothy Baldwin, Jey Han Lau, and Mark Stevenson. 2014. Representing topics labels for exploring digital libraries. In *Proceedings of the IEEE/ACM Joint Conference on Digital Libraries*.

Lukas Barth, Stephen G. Kobourov, and Sergey Pupyrev. 2014. Experimental comparison of semantic word clouds. In *Experimental Algorithms*. Springer.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

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

Jordan Boyd-Graber, David Mimno, and David Newman, 2014. *Care and Feeding of Topic Models: Problems, Diagnostics, and Improvements*. CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton, Florida.

Allison June Barlow Chaney and David M. Blei. 2012. Visualizing topic models. In *International Conference on Weblogs and Social Media*.

Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of Advances in Neural Information Processing Systems*.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Jason Chuang, Christopher D. Manning, and Jeffrey Heer. 2012. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*.

Jacob Eisenstein, Duen Horng Chau, Aniket Kittur, and Eric Xing. 2012. TopicViz: Interactive topic exploration in document collections. In *International Conference on Human Factors in Computing Systems*.

Thomas M.J. Fruchterman and Edward M. Reingold. 1991. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164.

Matthew J. Gardner, Joshua Lutes, Jeff Lund, Josh Hansen, Dan Walker, Eric Ringger, and Kevin Seppi. 2010. The Topic Browser: An interactive tool for browsing topic models. In *Proceedings of the NIPS Workshop on Challenges of Data Visualization*.

Ramanathan Guha and Rob McCool. 2003. TAP: A semantic web platform. *Computer Networks*, 42(5):557–577.

Matthew Hoffman, David M. Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347.

Andreas Hotho, Andreas Nürnberger, and Gerhard Paass. 2005. A brief survey of text mining. *Journal for Computational Linguistics and Language Technology*, 20(1):19–62.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. 2014. Interactive topic modeling. *Machine Learning*, 95(3):423–469.

Fred Jelinek, Robert L. Mercer, Lalit R. Bahl, and James K. Baker. 1977. Perplexity–a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Lauren F. Klein, Jacob Eisenstein, and Iris Sun. 2015. Exploratory thematic analysis for digitized archival collections. *Digital Scholarship in the Humanities*.

Jon Kleinberg. 2003. An impossibility theorem for clustering. In *Proceedings of Advances in Neural Information Processing Systems*.

Jey Han Lau, David Newman, Sarvnaz Karimi, and Timothy Baldwin. 2010. Best topic word selection for topic labelling. In *Proceedings of the Association for Computational Linguistics*.

Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. 2011. Automatic labelling of topic models. In *Proceedings of the Association for Computational Linguistics*.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.

Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. 2007. Automatic labeling of multinomial topic models. In *Knowledge Discovery and Data Mining*.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Andrew Mueller. 2012. Word cloud. `https://github.com/amueller/word_cloud`.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010a. Automatic evaluation of topic coherence. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010b. Evaluating topic models for digital libraries. In *Proceedings of the IEEE/ACM Joint Conference on Digital Libraries*.

Viet-An Nguyen, Yuening Hu, Jordan Boyd-Graber, and Philip Resnik. 2013. Argviz: Interactive visualization of topic dynamics in multi-party conversations. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Daniel Ramage, Susan T. Dumais, and Daniel J. Liebling. 2010. Characterizing microblogs with topic models. In *International Conference on Weblogs and Social Media*.

Patrick Riehmann, Manfred Hanfler, and Bernd Froehlich. 2005. Interactive Sankey diagrams. In *IEEE Symposium on Information Visualization*.

Eduarda Mendes Rodrigues, Natasa Milic-Frayling, Marc Smith, Ben Shneiderman, and Derek Hansen. 2011. Group-in-a-box layout for multi-faceted analysis of communities. In *Proceedings of the IEEE Conference on Social Computing*.

Evan Sandhaus. 2008. The New York Times annotated corpus LDC2008T19. *Linguistic Data Consortium, Philadelphia*.

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

Carson Sievert and Kenneth E. Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*.

Alison Smith, Jason Chuang, Yuening Hu, Jordan Boyd-Graber, and Leah Findlater. 2014. Concurrent visualization of relationships between words and topics in topic models. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*.

Alison Smith, Sana Malik, and Ben Shneiderman. 2015. Visual analysis of topical evolution in unstructured text: Design and evaluation of TopicFlow. In *Applications of Social Media and Social Network Analysis*.

Alexander Smola and Shravan Narayanamurthy. 2010. An architecture for parallel topic models. In *Proceedings of the VLDB Endowment*.

Hanna Wallach, David Mimno, and Andrew McCallum. 2009a. Rethinking LDA: Why priors matter. In *Proceedings of Advances in Neural Information Processing Systems*.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009b. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Knowledge Discovery and Data Mining*.

Ji Soo Yi, Rachel Melton, John Stasko, and Julie A. Jacko. 2005. Dust & magnet: Multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256.

Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad Alkhouja. 2012. Mr. LDA: A flexible large scale topic modeling package using variational inference in MapReduce. In *Proceedings of the ACM Conference on World Wide Web*.