Task-Driven Comparison of Topic Models

Eric Alexander and Michael Gleicher, Member, IEEE

Abstract—Topic modeling, a method of statistically extracting thematic content from a large collection of texts, is used for a wide variety of tasks within text analysis. Though there are a growing number of tools and techniques for exploring single models, comparisons between models are generally reduced to a small set of numerical metrics. These metrics may or may not reflect a model's performance on the analyst's intended task, and can therefore be insufficient to diagnose what causes differences between models. In this paper, we explore task-centric topic model comparison, considering how we can both provide detail for a more nuanced understanding of differences and address the wealth of tasks for which topic models are used. We derive comparison tasks from single-model uses of topic models, which predominantly fall into the categories of understanding topics, understanding similarity, and understanding change. Finally, we provide several visualization techniques that facilitate these tasks, including buddy plots, which combine color and position encodings to allow analysts to readily view changes in document similarity.

Index Terms—Text visualization, topic modeling

1 INTRODUCTION

Probabilistic topic modeling is an increasingly popular method for performing text analysis in a wide variety of domains. This suite of algorithms extracts collections of words that tend to appear in the same documents in an attempt to discover semantic connections. The ability to find such connections within large corpora without needing to read through every document makes topic models attractive. Due to the size and complexity of the resulting models, visualization can be helpful for exploring these results. There have been many tools in recent years designed to expose a single model to the user. However, these tools rarely facilitate direct comparison between models.

The ability to compare topic models offers a number of useful insights. For example, such comparison can help pick which model to use within the parameter space, which can be vast. (To be clear, we are using the term “topic model” to describe an instance of output given by a topic modeling algorithm run on a corpus.) Other reasons for comparison include validating findings (i.e., making sure that a pattern appearing in one model also appears in another) and evaluating different modeling or pre-processing techniques. Typically, these comparisons are relegated to numerical metrics, reducing comparison to the optimization of a single number. While this is useful for being able to iterate through many models quickly, these metrics can be inconsistent and often fail to align with human judgments [11, 30].

A single number also gives little insight into the significance of differences between models as it tends not to provide prescriptive recommendations beyond “this model ranks higher.” Difference is only significant to the degree that it affects what the user will actually do with the model. Therefore, we need to support task-driven topic model comparison. Such comparison would be easier to interpret, especially for users familiar with their tasks but unfamiliar with machine learning concepts like held-out likelihood and mutual information. By framing differences between models in terms of the task being performed, we can allow the user to decide which differences are significant for their particular use case. This can lead to a more nuanced understanding of the models, building user trust in the process.

Task-driven comparison must address the wide variety of tasks for which topic models are used. These range from full-corpus summary, to comparisons between documents or groups of documents, to investigations of evolution over time. To account for this range, we com-

Fig. 1: Buddy plots show consistency of document relationships across topic models by encoding similarity with respect to individual documents. In this figure, each row represents a document, with the rest of the corpus encoded as circular glyphs along the row. Distance from the row's document in one model is encoded using horizontal position, while distance in a second model is encoded using color. This combination of encodings lets us see similarities from two models within one row of glyphs. Deviations in similarity between the two models can be identified as breaks from a smooth gradient. Though the two models seem to correlate well with documents at either extreme (blue documents to the left, red documents to the right), we see dramatic shifts between different classifications for documents in between, identified by breaks in the blue-to-red gradient structure.
pose a list of comparative tasks that correspond to the most frequent single-model tasks, which we group into the categories of understanding topics, understanding similarity, and understanding change. We outline cross-model comparisons to be performed for each of these categories and describe visual techniques for doing so. In selecting the techniques seemingly best suited for the comparison tasks, we use a number of pre-existing encodings as well as a novel encoding for document similarity. Together, these enable in-depth exploratory comparison of topic models.

Our main contributions in this paper are:

- A characterization of the problem of topic model comparison, and an argument for a task-driven solution
- A survey of comparison tasks that corresponds to the single-model tasks for which topic models are already used
- Visual analytics approaches to address these comparison tasks, including the application of standard visualizations as well as more novel designs including buddy plots, a novel way of looking at relative distances within a corpus.

2 RELATED WORK

Topic modeling is a text processing technique that determines themes within a collection of texts using statistical analysis [6]. While there is a broad and evolving range of available techniques, most produce results of a similar form: topics are represented by sets of commonly occurring words, allowing documents to be assigned to topics by considering the words they contain. Most topic modeling techniques are probabilistic, creating weighted assignments of words to topics. Model comparison is typically done either by hand or using statistical metrics. The former method—analyzing each model individually using whatever tools are available—is often resorted to in practice, but is inefficient for large and complex models. Instead, statistical metrics of a model’s quality, such as predicted likelihood on a set of held out documents, provide more expedient methods of comparison. However, the variety of ways for calculating this likelihood tend to be inconsistent and inaccurate [32], often disagreeing with human judgments of the documents [9]. Chang et al. [9] offer an alternative method for computing model quality called “topic intrusion,” which asks human subjects to pick out topics that have been artificially inserted into document distributions. Though this can be effective, it relies on having a large pool of participants to perform evaluation.

Rather than judging the quality of a model as a whole, some metrics instead judge individual topic quality. These metrics have used pointwise mutual information computed with an external corpus [29] and document co-occurrence within the original corpus [26]. However, these may disproportionately favor certain modeling types [30]. AlSumait et al. [3] have developed a topic significance ranking that defines a topic distribution’s quality as its distance from three “junk” distributions: uniform across the vocabulary, equivalent to the corpus-wide distribution of words, and uniform across all documents in the corpus. While these are interesting and important qualities, the method sacrifices explanatory power for simplicity by combining them into a single numerical metric. These metrics have been shown to be inconsistent with each other and human judgment [11, 30].

Other metrics of topic quality take advantage of human input. “Word intrusion” metrics measure the consistency with which subjects can identify a fake topic word inserted into the list [9]. Chuang et al. [11] compare topics to expert-generated categories to filter them into classifications of junk, fused, missing, repeated, or resolved. However, though both of these methods are informative, they can again be impractical for those without access to a wide pool of participants. Some visual approaches to model evaluation and comparison have been developed. Chuang et al. [11] use matrix views to compare topics to ground-truth concepts generated by a group of experts. Though their method could be extensible to comparing two models, it is difficult to answer questions of why the models are different and which are better without the expert ground-truth. Crossno et al. [14] use bipartite graphs and two-dimensional embeddings to compare models built with LDA and LSA. Though their visual techniques are informative and use encodings very similar to those we present for aligning topics, they do not let the user drill deeply into individual relationships between topics, nor do they address the wider variety of tasks for which topic models are used. Our encodings for topic alignment add finer detail about different levels of alignment in the form of Pareto bar charts, and allow direct comparison of the words included so as to help identify split and merged topics across models (see Section 4.1.1).

3 MOTIVATION

There is no one perfect model for any given corpus. Rather, there are many possible models that can be created, each with different strengths and weaknesses depending on the scope of the user’s questions and the granularity of the patterns in which they are interested. In this section, we outline these decisions and tasks, and describe why we believe they require a different set of model comparisons than is currently available.

3.1 The parameter space

The decisions that go into creating a topic model form a vast parameter space. The first is deciding which technique to use. Latent Dirichlet Analysis (LDA) [6] is increasingly common, but other methods include latent semantic analysis (LSA) [16] and non-negative matrix factorization (NMF) [36]. Some versions incorporate expert knowledge [5] or build hierarchical topic structures [19]. These model types have been shown to perform differently depending on the task [10, 14, 30].

There are a number of technique-specific parameters that affect the specific instance of a model. The most obvious is the number of topics to extract, the optimum for which will be highly dependent on the corpus, but there are often many others. For instance, in LDA, the coarseness of the topic and document distributions are controlled by two hyperparameters, \( \alpha \) and \( \beta \). Although end-user topic modeling packages like MALLET often hide these parameters from the user [25], they can have a significant impact on topic legibility [31].

Though there has been some work visually exploring this hyperparameter space [11], there are also many parameters that go into the pre-processing steps of model creation. For example, the analyst may choose to filter out stopwords: words that are considered to be without semantic meaning but may still be common enough to dominate statistical analysis. Obvious examples include articles, pronouns, and other function words, but can also include proper nouns or domain-specific terminology. Though there are default lists built into many tools, the precise definition of a stopword largely depends on what the user is looking for. For example, for a journalist building a topic model on newspaper articles, proper nouns like “Obama” might be a semantically important to include in analysis, while a literature scholar might not want to associate two plays just because they both contain a character named “Antonio.” Even treatment of function words can vary, as there are scholars interested in such words’ ability to distinguish between groups or authors [21].

Another pre-processing step that is not always considered in evaluation is the practice of “chunking” documents. When working with longer documents, the concepts of context and co-occurrence can get a bit blurry. We might be able to assume that two words in a journal abstract are semantically related, but what about two words at either end of a novel? Or a chapter? Or a scene? A common way of managing this discrepancy is to cut larger documents into chunks—either semantically (e.g. chapter or scene) or numerically (e.g. by number of words). However, the size of these chunks is largely dependent on the level of granularity at which the user wants to explore patterns—again, on the particular task they have in mind.

There are many other possible parameters a user might want to compare (see Section 5). Though there are some objective performance differences, given the task specificity of these parameters, it is not enough to rely solely on statistics that return a number. We want to instead help the user build insight into these decisions by supporting comparison that is parameter agnostic, showing the effects of their choices on the tasks they want to perform.
### Table 1: Mapping Single Model Tasks to Comparison Tasks

<table>
<thead>
<tr>
<th>Single model task</th>
<th>Model comparison task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding topics</td>
<td>Topic alignment</td>
</tr>
<tr>
<td>Understanding similarity</td>
<td>Distance comparison</td>
</tr>
<tr>
<td>Understanding change</td>
<td>Timeline comparison</td>
</tr>
</tbody>
</table>

#### 3.2 The tasks

Understanding the tasks most often performed using topic models can help inform decisions within this parameter space. From our own experience and examination of literature, we believe that the majority of uses fall into these categories:

**Understanding topics** Topic models are often used as a tool for summarizing documents, pulling out the most important topics without having to read through everything. Connected to this task is understanding, summarizing, and naming individual topics. Though topics are often represented by their top three words [8], word clouds [35], or ordered lists [12], how best to convey them comprehensively remains an open research question.

**Understanding similarity** Topic models can provide an unsupervised distance function for comparing documents, both globally (how do documents cluster across the corpus; how well do clusters match human categories) and locally (which documents are most similar to a particular document of interest). Using models as a metric for similarity can also be useful for identifying individual documents that are exemplary of a topic or trend, or else that are outliers from the rest of the corpus [2, 13].

**Understanding change** Seeing how corpora change over time is one of the most common uses of topic modeling. Temporally focused tasks include both looking for large trends and patterns as well as particular events [22], often in correlation with outside historical data. Visual analysis is particularly well-suited to address these sorts of tasks, and many tools incorporate some form of temporal river flow visualization for doing so [15, 20, 34].

Each of these single-model tasks corresponds to a type of comparison that a user might make between models (Table 1). The primary comparison task involved in understanding topics is **topic alignment**: how well do the topics from the two models match? If they do not, why not, and what do those differences mean?

When considering the similarity of documents, topic models can be used to calculate distances between documents which can be compared. These **distance comparisons** can be made both globally (do the models agree upon how to group the documents) and locally (how do the document neighborhoods change from one model to another). Finally, we refer to temporal comparisons between a collection of documents as **timeline comparison**. How well are the models aligned in time? Do they pick out similar important events? Do topics that share similar words also evolve in the same ways?

There is overlap between these comparative tasks, but the requirements of these tasks are sufficiently distinct to require different visual techniques to conduct effectively.

### 4 Visualizing comparison

Most topic models can be represented as three matrices containing three different sets of probability distributions: a matrix associating documents and topics, a matrix associating topics and words, and a matrix associating documents and words. One could frame the problem of topic model comparison as simply one of matrix comparison. However, though matrix views can be useful, there are a number of reasons why they are insufficient by themselves. It is difficult for visualizations of these matrices to scale, with topics on the order of hundreds, documents on the order of hundreds or thousands, and words on the order of tens of thousands. Depending on the questions the user might ask, pure matrix views might not give much flexibility for exploring individual documents or derived relationships between smaller sets of documents and topics. Finally, these matrices do not connect directly to the tasks.

Given the importance of such relationships and metadata in the tasks for which models are used, we explore visual encodings that address them directly in the following sections.

#### 4.1 Understanding topics

Our predominant concern when considering topics across two models is **topic alignment**: which topics best match with one another, and how. Matching can mean either sharing the same documents or the same words. This comparison is important for tracking consistency and correspondence across models. In two closely aligned models, we would expect to see a one-to-one matching, but this is not always the case, either because of parameter differences or random variations.

There are a few common phenomena to look for when comparing topics across models:

- **Matched topics** - Some topics will be (close to) direct matches of one other, sharing distributions of both words and documents.
- **Split/merged topics** - A topic from one model will occasionally split into multiple topics in another model (or, multiple topics may merge, depending on the direction of comparison). This happens often with—but is not limited to—comparisons between models with different numbers of topics.
- **Absent topics** - Finally, some topics from one model will simply have no correlated counterpart in the other model.

Unfortunately, identifying these phenomena only gets us so far. Without a ground truth to compare against, the user must be able to drill into these splits and merges to understand whether they are semantic or random, and what to do about them. To this end, we must not only support topic alignment, but topic comparison as well.

##### 4.1.1 Topic alignment

The alignment between two sets of topics is represented by a matrix of the distances between each topic in one set and the other. These distances can be computed as the vector distances of either the words in the topic or the documents containing the topic, by any appropriate vector distance transform (typically cosine distance). We convert these distance matrices to similarity matrices to facilitate matrix operations like re-ordering and sparsification.

Given the topic similarity matrix, it is easy to identify each topic’s corresponding best match as the maximal element of its corresponding row or column. However, more information is typically desirable. For example, in examining a match, it is useful to know how strong the match is (the topic similarity), whether the match is “clean” (i.e. is the best match clearly better than the alternatives), and whether the match is bi-directional (i.e. that the best matching topic doesn’t have a better match for itself).

Since topics are categorical, we can re-order them to best expose the matching structure. While a number of potential algorithms are possible (see [27] for a survey), we find that a simple sparse matrix diagonalizer is sufficient as we are generally concerned with only the primary diagonal structure (the few best matches for each). Our approach first sparsifies the similarity matrix, reducing the number of non-zero elements such that each row and column has a minimum (typically 3 or 5) of non-zero elements. We then use an asymmetric variant of the Cuthill-McKee algorithm (greedy breadth first search) to optimally order the rows and columns. This provides a matrix where the strongest matches appear close to the diagonal.

We provide a heatmap view of this re-ordered similarity matrix (see Figure 2). We also provide a standard “parallel coordinates” node-link view, showing links between similar topics. We use colors on the links to encode the match type: green for a bidirectional best match, purple for a one-way best match (with tapered lines indicating directionality), and less salient grays to encode other links. Weak matches can be...
Once the user has identified instances of split, merged, or pseudo-matched topics, the question becomes what to do about it. Is the split semantic or arbitrary? To find out, the user needs to see the words. The problem is that there are potentially hundreds or thousands of words in the topics, so simply reading through the lists is not feasible. One approach would be to limit the user to seeing only the top-most words in each model, but the other words do have an effect (as we see in Section 5.4), so we don’t want to throw them out completely.

We use a color-field rank comparison visualization to allow the user to see patterns across the entire topic lengths while also being able to drill into the individual words (see Figure 3). The visualization takes an object topic from one model and potentially multiple reference topics from the other model. For each reference topic, we create a color transfer function encoding words with colors from a continuous ramp based on their rank within the topic. These ramps are interpolated from discrete ColorBrewer ramps [7]. We then use these reference ramps to show where each reference topic’s words appear in the object topic, where they overlap, and where the object topic is unique.

This makes it easy to spot instances of split topics, as shown in Figure 3. To evaluate these splits, we have a moveable window allowing users to zoom into the individual words. For example, in Figure 3, the red topic’s words seem to be associated with wealth while the blue topic seems more about family. This looks like a semantic split, so the user may favor the model containing the split topics rather than their merged version.

4.2 Understanding similarity

Using a topic model to calculate similarities between documents provides insight into specific relationships within a corpus. Similarity can be both a local concept (which documents are most similar to a given reference) and a global one (how do documents cluster together). By treating document distributions as vectors, we can compute distance matrices to look at such similarities. Though cosine distance is the most frequently used, other distance metrics include KL-divergence [3] and rescaled dot product [11].

In this section, we look at a novel method for viewing document similarities, as well as a method for comparing how those similarities can be used to cluster documents in different models.
4.2.1 Buddy plots

A common way of looking at document distance is to place documents into a two-dimensional embedding [14]. When looking at change in similarities between models, we can consider how documents move within the embedding. As distances in topic models are a product of dimensionality reduction, they have no semantic meaning. Therefore, apparently large changes in position from one embedding to the next may not correspond to any meaningful difference, and it is difficult to know which pairwise relationships are authentic as opposed to artifacts of the dimensionality reduction.

One way of seeing corpus movement that sidesteps these challenges is to shift our vantage point from an overhead, all-inclusive one to that of a single document. To see how similarities have changed for this document, we can effectively hold it stationary across two models and observe how the rest of the corpus shifts around it. This shift in vantage point can give us local insight around the stationary document. Doing so for multiple documents (and potentially juxtaposing their respective vantage points together) can give us a more global sense of the changing similarities across the two models. We have created visualizations called buddy plots that give a view of the corpus from this shifted perspective.

Buddy plots are customizable and can map a variety of data to different encodings. They show topic similarity from the perspective of a single reference document. The rest of the documents (the reference’s “buddies”) are arranged linearly along a horizontal axis based on their distance from the reference document. When comparing models, we can combine the sets of distances in one of two ways:

**Single buddy plots** have only one axis, and by default encode one set of distances as horizontal position and the other as color along a diverging ramp (Figure 4a).

**Parallel buddy plots** have two axes, typically encoding each set of distances on its own axis and connecting them with parallel-coordinate-style edges (Figure 4b).

These two sets of encodings create a trade-off between scalability and accuracy. Single buddy plots fit more data into a tight space, and allow us to leverage perceptual gradient detection to make judgments about document movement. If the two sets of distances agree, the documents should form a smooth gradient from left to right. Departures from this gradient become pre-attentively identifiable [1], drawing the user’s attention to outliers that can be inspected further or explicitly zoomed in on (see Figure 7). Parallel buddy plots, on the other hand, give more detailed information about precisely which documents are moving where across the two models by giving each model its own axis and drawing explicit connections. By default, color remains consistent across both axes (though this is redundant with position) to facilitate easy tracking of movement and direct comparison.

Globally, combining buddy plots gives a sense of how similarity changes across the corpus between the two models. With single buddy plots, deviations from the gradient can give a sense of the cross-model consistency (Figure 1). Patterns of parallel buddy plots can show directional or density trends from one model to the next (Figure 14). Though single buddy plots are more space efficient, it is still difficult to fit an entire corpus onto the screen. As has been seen elsewhere [2], reordering the documents by task is a simple but effective way of bringing the most important items into view. We offer a variety of metrics by which to reorder documents based on what the user is most interested in, such as the average change in distance or rank for documents (i.e. how much a document moves with respect to the rest of the corpus).

While being able to see movement within the entire corpus can be informative, many tasks are only concerned with exploring the most similar documents. As such, we want to give users a sense of how these few documents—the reference document’s local neighborhood—change from one model to the next. Neighborhoods can be thought as the closest documents either by distance (e.g. documents within .1 of the reference document) or by rank (e.g. the top 10 closest documents to the reference document). To measure change in neighborhoods, we devised a metric called pareto radii. Pareto radii indicate how much a document’s neighborhood has spread out from one model to the next (for multiple definitions of neighborhood). Multiple radii can indicate increasing proportions of the neighborhood. Specifically, for distance neighborhoods, when looking for change from model A to B within the neighborhood of some reference document k, the pareto radius \( r_{dist}(d, p) \) is the smallest radius around document \( k \) in model \( B \) that contains the proportion \( p \) of all documents within distance \( d \) of document \( k \) in model \( A \). Pareto radii can also be defined for rank neighborhoods: when looking for change from model A to B within the neighborhood of some reference document \( k \), the pareto radius \( r_{rank}(r, p) \) is the smallest radius around document \( k \) in model \( B \) that...
contains the proportion $p$ of the top $n$ documents of document $k$ in model $A$. These radii can be encoded using a gradient for increasing values of $p$ as seen in Figure 5. Length of pareto radii can also be an effective encoding for sorting buddy plots, bringing those with the most or least movement within their local neighborhoods to the top for user inspection.

Through flexibility in the encodings, buddy plots can be tuned to answer a variety of different questions (see Figure 6). In addition to distance, document color can be used to encode rank, change in distance or rank across the two models, or metadata about the documents (like genre or domain). Edge color in parallel buddy plots can be changed to highlight documents that move far, stay consistent, or move in a particular direction relative to the reference document. Also, the draw order of both documents and edges, like zooming, can help combat overdraw for corpora with many or tightly clustered documents. The default draw order keeps documents that were closer in one model on top, making it easier to pick out movement within a document’s relative neighborhood (see Figure 7). Other potential draw orders include rank within either model, as well as change in rank or distance.

**Fig. 5:** Pareto radii are used to indicate how much a document’s “neighborhood” spreads out from one model to another. Here, the gray gradient shows pareto radii in a trimmed model built on AP news documents that are needed to cover 20, 40, 60, 80, and 100 percent of the closest 20 documents to the reference document in the untrimmed version of the model.

Through flexibility in the encodings, buddy plots can be tuned to answer a variety of different questions (see Figure 6). In addition to distance, document color can be used to encode rank, change in distance or rank across the two models, or metadata about the documents (like genre or domain). Edge color in parallel buddy plots can be changed to highlight documents that move far, stay consistent, or move in a particular direction relative to the reference document. Also, the draw order of both documents and edges, like zooming, can help combat overdraw for corpora with many or tightly clustered documents. The default draw order keeps documents that were closer in one model on top, making it easier to pick out movement within a document’s relative neighborhood (see Figure 7). Other potential draw orders include rank within either model, as well as change in rank or distance.

**Fig. 6:** Buddy plots allow researchers to explore comparisons by combining a variety of encodings and data. Though most data and encodings can be combined, we have a set of defaults that seem well suited to single and parallel buddy plots.

4.2.2 Comparison of clustering from models

The distance metrics provided by topic models are often used to cluster groups of related documents. Users concerned with such grouping tasks may want to compare models based on their performance at clustering documents, considering that different models may induce different groups. It is possible to use each model to cluster the documents, and then to compare the resulting clusterings using any number of clustering comparison tools [18]. Such clustering comparison is no different from comparing two different clusterings created from the same model. This emphasizes a key point: since there are many possible clusterings that may result form the same model, a difference between clusterings created by two different models might just be a matter of adjusting some of the clustering parameters.

For understanding how differences in models affect clustering, we introduce a method that avoids considering a single clustering, but instead tries to help assess the intrinsic clustering ability of a model. Rather than performing clustering once, we repeat the process many times. This allows for the non-deterministic algorithm finding different solutions, as well for many different settings of clustering parameters. We then provide an assessment of how consistent this set of clusterings is, and allow the user to compare these assessments. We visualize this set with a matrix that counts for each possible pair of documents how often the documents appear in the same cluster. The intuition is that if a model is good for clustering, then documents should consistently either be in the same cluster or not, even as the parameters are changed. We display these matrices as heatmaps, ordering the rows using the Cuthill-McKee algorithm to best expose structure. An example comparing two models is shown in Figure 8. In this simple example, multiple runs of K-means clustering is run on each model. Because of the non-determinism of the starting points, the results are different on each trial. However, with the 10-topic model, the clusters are more consistent, suggesting that the boundaries in this model are clearer and less likely to simply be artifacts of the clustering process.

4.3 Understanding change

Seeing how content changes over time is a very prevalent use of topic models. To this end, a common visualization technique is to create “river flow” style diagrams to display these changes [34, 20, 15]. Though there are many variations, such diagrams typically have individual colored bands for each topic. The width of these bands encode
the portion of the corpus represented by a particular topic at a point in time (with time represented as position along the horizontal axis).

For a user interested in exploring such temporal patterns, it may be important to see whether different events or timepoints are highlighted differently by two models. To this end, we have created asymmetrical topic flow diagrams to compare trends over time across models (see Figure 9). Topic densities from the two models are plotted along a horizontal time axis, with one model’s topics appearing above the axis and the other model’s topics appearing below. This allows the user to look for symmetry (or asymmetry) in the trends and events being highlighted by the two models.

Especially for models built on the same documents, there may not be large changes to the global shape of the topics. However, the behavior of individual topics can exhibit interesting differences to lead a user to prefer one model over another. To explore such differences, we allow users to directly compare aligned topics across the two models. Brushing over an individual topic in one model will highlight its most aligned topics in the other model (using the same topic alignment method as described in Section 4.1.1). Clicking on the topic will filter away all other topics but the selected one and its closest matches. This can provide an informative comparison between split/merged topics. Such topics may be split uniformly across time, or else might alternate, highlighting different events (see Figure 10). This distinction can help the user decide whether the split is a semantic one or a random artifact of the model parameters.

5 Usage Scenarios

In this section, we describe a number of usage scenarios that illustrate the capabilities of our comparison techniques on real data. Some of these comparisons fall into the traditional motivations for model evaluation (parameter optimization, etc.), while others would not be possible with a statistical metric.

These comparisons use models built on three separate corpora: the 36 plays of William Shakespeare, a collection of 1127 abstracts from select IEEE sponsored visualization conferences from 2007-2013 (including SciVis, InfoVis, VAST, BioVis, and PacificVis), and a collection of 2250 articles from the Associated Press [33]. All models were built with MALLET [25].

5.1 Same parameters (Visualization Abstracts)

Topic models are inherently probabilistic, and there has been concern that some (like LDA) can be inconsistent across different runs with the same parameters [10, 24]. A researcher wanting to validate their findings in one model may want to confirm that there are no significant changes due to chance or hidden parameters, and that any existing changes do not affect the patterns in which they are most interested.

We decided to investigate this phenomenon. To do so, we built two 25-topic models on our collection of visualization abstracts, using the default parameters for $\alpha$ and $\beta$. We found some inconsistencies, which might disproportionately affect different use cases.

Specifically, we saw change in both the topic alignment and document similarity. Though the differences in topics may fall within a reasonable allowance (see Figure 13a), changes in document similarity tended to be much more dramatic (see Figure 11). Though using MALLET’s built in hyperparameter optimization improved the topic alignment (see Figure 13b), document neighborhoods remained inconsistent. This suggests that while LDA may work well for topic-based tasks like corpus summarization, it may not be as well suited for tasks that require consistent evaluation of similarity.

5.2 Different numbers of topics (Shakespeare)

As mentioned in Section 2, the most important parameter for building a topic model is the number of topics. Too low of a number will result in overly broad topics or merges between unrelated topics, while too high of a number can create non-semantic splits within topics and hide larger trends. To observe these differences, we built models of 10 and 50 topics on the works of William Shakespeare.

The primary difference we saw between the two models was how they were clustering. Figure 8a shows a heatmap of the consistency of clustering across 20 runs of k-means upon the 10-topic model. We can see that with the minor exception of the cluster in the direct center (and in particular, Romeo and Juliet), the groups being formed are
5.3 Document chunking (Shakespeare)

As described in Section 2, longer documents can muddy the concept of document co-occurrence. Should words still be considered to be co-occurring if they occur in different paragraphs, or pages, or chapters? Intuition would say that longer chunks will create more general models, but it can be hard to predict the effects a priori.

We built two models of 25 topics on the works of Shakespeare: one in which the documents were divided into 1000-word chunks (as recommended in [23]) and stitched back together after being tagged by the model, and one in which the documents were treated as whole.

A quick glance at the buddy plots showed that there did not seem to be dramatic changes in document neighborhoods, as the blue-to-red gradients were relatively maintained. Expanding the rows out into parallel buddy plots exposed a consistent trend: though the order of document distances does not change much, documents seemed to grow much further apart in the non-chunked model (as seen in Figure 14). This makes sense as the process of stitching small chunks back together would tend to create document vectors that cover a wider range of topics. The fact that relative orders do not change much may suggest that we can trust relationships we find in the un-chunked (and correspondingly much faster to build) model.

There are a few salient red edges, however, showing that some documents do move dramatically against the grain. For one such example, in Figure 4b, _Taming of the Shrew_ grows much closer to _Comedy of Errors_ in the unchunked model, while everything else moves further away. Looking at the bipartite topic alignment of topics specifically contained within these two documents, we noticed that there is one topic that they share in the unchunked model that is best aligned with two topics in the chunked model that they distinctly do not share.

Drilling into the words shows that this is a semantic split (see Figure 3), with the red topic dealing with wealth and the blue topic dealing more with family. Depending on the user's interest in these specific topics and documents, incorporating this split may be worth the added time and effort of the chunking process.

5.4 Trimmed topic models (AP articles)

One of the problems with using topic models for text analysis is that they can be difficult to understand for a lay audience. Among the challenging concepts for users without a background in statistics or machine learning are the existence of overlapping topics (i.e. words in multiple topics), the distribution of low-frequency words, and the probabilistic nature of topic tags. In attempts to make topic models easier to understand, we have considered a variety of ways of simplifying the distributions created by algorithms like LDA.

One such method is to simply trim away words from the bottom of the topic distributions. This certainly makes the topics simpler, and reduces them to something more closely matching their visual representations to users, which tend to be lists or word clouds of the top most frequent words. We were interested in looking at how much such an operation would affect the relationships between documents within a corpus. If it had no effect, then perhaps trimming in this fashion would be a useful way of greatly simplifying topic representations.

We built a model of 50 topics on a corpus of articles from the Associated Press [33], trimmed the bottom proportions at a number of different increments, and looked for the changes in document relationships using buddy plots. There was rather dramatic movement from the untrimmed model to the trimmed versions. Figure 1 shows a subset of single buddy plots comparing the untrimmed model to a model with topics trimmed down to the 50 most frequent words. Though we still see a semblance of a blue-to-red gradient, the darkest blues (the original document neighborhoods) are spread out dramatically. Some documents have orange and red documents bleeding far to the left. By drilling into a parallel view of one of these documents (Figure 12), we can see that the documents become intensely jumbled.

**Fig. 11**: This parallel buddy plot compares distances between two 25-topic models built using the same (default) parameters on a corpus of 1127 visualization abstracts. Color encodes distance in the first model for glyphs along both axes for consistency, and edge lines are ordered by greatest magnitude of change in distance. We can see that there has been a dramatic shift in similar documents across the two models.

**Fig. 12**: This parallel buddy plot compares distances from a 50-topic model built on a corpus of Associated Press documents to distances from that same model when all but the top 50 most frequent words are stripped from each topic. (Color encodes distance in the first model and edges are drawn in increasing order of change in distance.) We see that the act of stripping words from the topics creates a very dramatic change in the document's relationships with the rest of the corpus.

![Fig. 13](image-url) 

(a) This heatmap shows the alignment from two models generated using MALLET's default parameters. Though there is a strong diagonal, other strong matches may indicate split topics.

(b) This heatmap shows two models that were also generated using the same parameters, but with MALLET's hyperparameter optimization. This seems to improve the topic alignment.

**Fig. 13**: These topic alignment heatmaps show comparisons of two pairs of topic models (25 topics on a corpus of visualization abstracts) generated using the exact same parameters.
concepts and words. It can be easy to forget that a topic is more than just its top 10 words. This exploration makes us wary of representations that hide too much of the full size and complexity of topics. We hope to explore differences created by other methods of simplifying or sparsifying topic models.

6 Conclusion

In this paper, we have argued for the importance of explicit, task-driven topic model comparison. We have offered visual techniques for making comparisons associated with the tasks of using topic models to understand topics, similarity, and change within text corpora. Among these techniques, we have introduced buddy plots, a novel visualization for viewing changes across two sets of distances by restricting our vantage point to that of a single reference document. Finally, we have described a number of use cases that exhibit the variety of comparisons that can be made with these methods.

We plan to combine these approaches into a fully deployed, interactive tool. There are a number of technical improvements that can be made to our algorithms (e.g. considering fuzzy matches in topic alignment). However, we also see a number of future directions for work in task-driven model comparison more generally. Though comparison between two models is important for understanding differences, such a restricted focus can also be a limitation for exploring the large decision space. We believe that there can be a benefit to combining pairwise comparison with higher-level overviews that incorporate more models. Tied to this, we would like to introduce better aggregation to combat the issue of scale, as our current implementations can break down with corpora in the thousands.

Though we believe that our three categories of understanding topics, similarity, and change cover the vast majority of topic model uses, there are a handful of other tasks we would like to support. For instance, there is a growing number of users who use topic models not to save them reading, but to direct their reading, augmenting individual words and passages with topic data [2]. We would like to be able to enable comparison at this level, informing users of what different models show about the passages and flow of individual documents.

We also want to provide tools to allow users to connect differences observed between models to actionable insights upon how to further improve and tune their models. Some work has been done on actively tuning models by merging or breaking up topics [10], pulling documents apart or closer together [17], or directly intervening in the ongoing iterations of the algorithm [28, 4]. It may be helpful to allow users to tune not just single models, but perhaps to pull the best parts of a variety of models together. We believe that the sorts of comparisons outlined in this paper would be useful in identifying which aspects to separate or combine.

Finally, we are interested in seeing these techniques applied to a wider variety of corpora and domains. We have only scratched the surface of potential comparisons to be made, and hope to use these methods to learn more about things like stopword usage, different modeling types, and models built on non-intersecting sets of documents.

Acknowledgments

This work was supported in part by NSF award IIS-1162037 and a grant from the Andrew W. Mellon Foundation.

References


