**Enabling Exploration and Hypothesis Formation within Topic Models**

by

Eric Carlson Alexander

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2016

Date of final oral examination: 11/17/2016

The dissertation is approved by the following members of the Final Oral Committee:
Michael Gleicher (Chair), Professor, Computer Sciences
Jerry Zhu, Professor, Computer Sciences
Mark Craven, Professor, Computer Sciences & Biostatistics
Mark Vareschi, Assistant Professor, English
Michael Witmore, Director, Folger Shakespeare Library

*To Anna. Let the adventures never end.*

## ACKNOWLEDGMENTS

*I can no other answer make but thanks, and thanks, and ever thanks*

— Sebastian, *Twelfth Night*

This dissertation was made possible thanks to the help and support of many people. I first want to thank my committee for their guidance and feedback over the course of my graduate career. Most especially, I want to thank my advisor, Mike Gleicher, who gave me the opportunity to combine multiple passions into my work, supported me as I learned the ropes, and helped me grow as a researcher, writer, and speaker.

I would like to thank my collaborators in the Visualizing English Print project: Mike Witmore for his never ending enthusiasm and encouragement; Jonathan Hope for his great ideas and dry wit; Robin Valenza for her support and creativity (and for coming up with such great names!); Heather Froehlich for fantastic feedback and even better introductions; Deidre Stuffer for more help than I can ever repay; Erin Winter for not throttling me after having to work with my code. Thank you to Bilge Mutlu and Steve Franconeri for helping me understand proper experimental design. I would like to thank all of my co-authors: Mike Gleicher, who collaborated on Chapters 3-6; Joe Kohlmann, Robin Valenza, and Mike Witmore, who collaborated on Chapter 3; and Chih-Ching Chang, Mariana Shimabukuro, Steve Franconeri, and Chris Collins who collaborated on Chapter 6.

I would like to thank my labmates in the Visual Computing Lab: Danielle Albers Szafir, Michael Correll, and Alper Sarikaya for helping me navigate the world of visualization; Brandon Smith and Sean Andrist for always being down for another cup of coffee; Nathan Mitchell for his patience, technical expertise, and always being game to argue and laugh about nothing.

I would like to thank my mentors at Carleton College, especially David Liben-Nowell for being willing to don many hats to help me navigate uncharted waters.

I would like to thank my friends and family for supporting me during my time in graduate school. Most of all I would like to thank my parents. Thank you for not pressuring me to join the family business, and then being endlessly supportive when I found my way there on my own.

Finally, thank you to Anna, for the reinforcement, for the laughs, for the ab workouts, for the late night sanity checks, for all of the adventures, and for making graduate school into the absolute best years of my life, so far.

The work in this dissertation was funded by NSF award IIS-1162037 and a grant from the Andrew W. Mellon Foundation.

CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## ABSTRACT

Text is ubiquitous, especially in the digital form. From websites, to news articles, to academic publications, to literature, the amount of text that is available for analysis has grown far beyond what researchers can make sense of unaided. Statistical models of text help researchers gain insight into large corpora that would be impossible to achieve through manual inspection of documents alone. However, such mathematical models can be difficult for researchers to make sense of, especially for those without statistical expertise. Affording visual exploration of these models can make them accessible and comprehensible to researchers in a wide variety of domains.

In this dissertation, I describe task-driven, visual exploration of probabilistic topic models: a class of text models that extract collections of words appearing together within a corpus. Specifically, I present an approach that helps researchers not only observe trends and patterns within documents, but also form *explanations* of those trends by drawing connections between them and the underlying data. I identify techniques for visual exploration of a single model and visual comparison of different topic models. I embody these techniques in a system that combines the analytic practices of close and distant reading to help researchers form and evaluate hypotheses about the documents. In addition to describing use cases carried out with domain collaborators using real data, I present experiments evaluating visual encodings used within the system. The main contributions of this dissertation are a task-driven approach for comparing and exploring document collections using topic models, a system embodying this approach, and evaluations of tagged text and word size encodings for use in such tasks.

# 1 INTRODUCTION

*If we offend, it is with our goodwill.*
*That you should think we come not to offend,*
*But with goodwill. To show our simple skill,*
*That is the true beginning of our end.*
*Consider, then, we come but in despite.*
*We do not come, as minding to content you,*
*Our true intent is. All for your delight*
*We are not here. That you should here repent you,*
*The actors are at hand, and, by their show,*
*You shall know all that you are like to know.*

— Peter Quince, *A Midsummer Night's Dream*

Text has long been ubiquitous in our culture. From the books, newspapers, and magazines we consciously peruse down to the signs and labels we take in almost imperceptibly, we are surrounded by the written word. In the modern age, this abundance of text has exploded even further as old media have been digitized and new media have arisen. With so much text available, it is natural for us to ask what we can learn from it all.

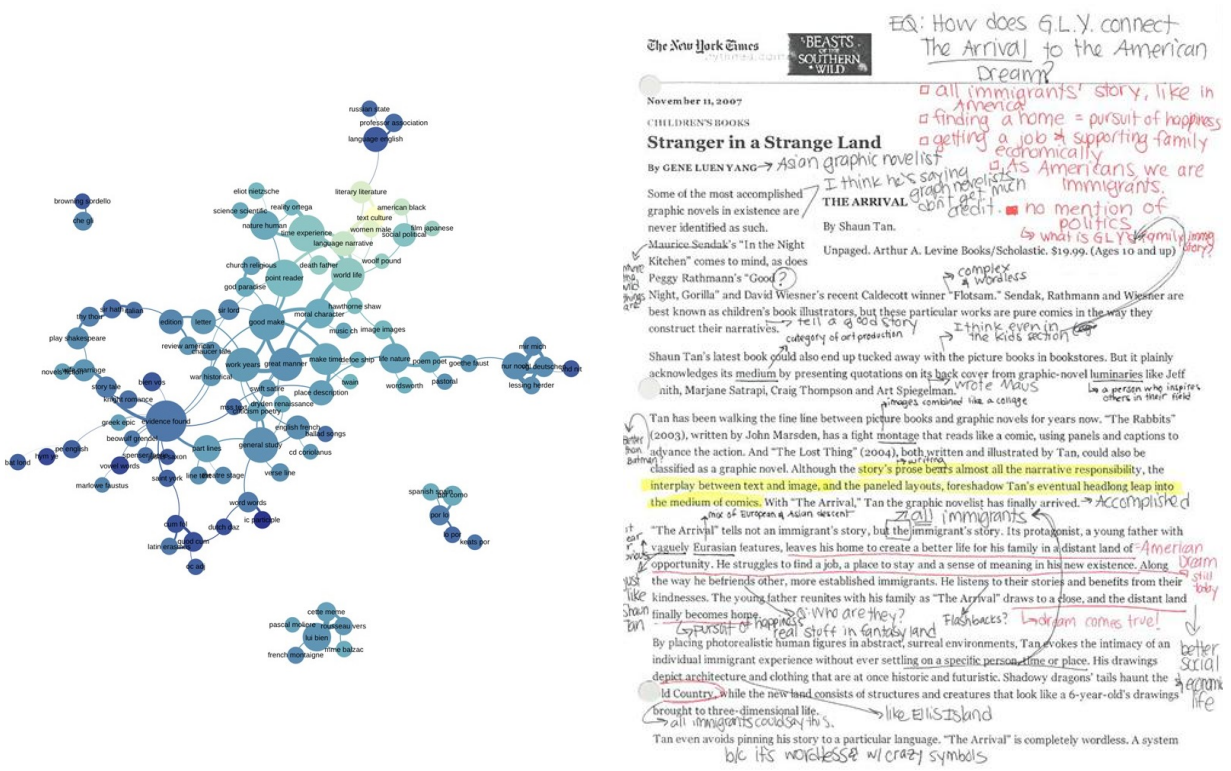
There are a number of challenges standing in the way of someone trying to pull insight from large amounts of text. The first challenge is one of scale. While large collections contain a wealth of information, they often also embody more text than any one person could hope to read unaided. This is where computational models of the documents can prove useful. What would take a person hours to read can be processed by a computer in fractions of a second. Books that are hundreds of thousands of words long can be transformed into vectors of numerical features which can then be aggregated to form summaries, clustered to find groups, or compared to find similarities and differences. Statistical models of text can help researchers discover representative patterns and trends that would have been impossible to find through manual inspection alone.

However, statistical modeling creates new challenges for textual research even as it addresses others. One such is that a statistical model adds an additional layer of complexity between the researcher and the ideas contained within the words of the documents. Unfortunately, researchers with the knowledge and background required to make sense of the original documents may not have the expertise to make sense of the statistical abstraction of the documents represented in the model. For example, a history scholar looking for patterns in a historical dataset may not feel comfortable making similarity calculations to see which documents are most closely related to a particular document of interest.

Data visualization can help fill this gap. Translating complex data into visual terms can make analysis more accessible for those without statistical expertise. Even regardless of expertise, our perceptual systems are incredibly powerful tools for recognizing patterns in large amounts of data, helping researchers once again battle the challenge of scale. As such, there are a growing number of visual analytics tools and techniques designed to help researchers understand statistical models of text. Such tools often present a "bird's eye view" of the documents, helping researchers to summarize them, find similarities, compare groups, or observe change over time.

However, when presented with such an observation (e.g., two similar documents,

a difference between groups, or a topic with increasing prevalence over time), the immediate next question is almost always "Why?" *Why* do we see this pattern? What is *causing* it? What are the attributes of the *underlying data* that make up the effect being observed in the statistical model? Ideally, the researcher would be able to form a potential explanation of the effect that is both articulable in the language of their domain as well as being an accurate reflection of the mathematics of the model. Without such an explanation—along with arguments showing how the data support or contradict the explanation—the "discovery" may seem empty. Unfortunately, visual tools that focus exclusively on abstract, aggregate views of the text do little to provide the low-level context required to form such hypotheses.




(a) Distant Reading (b) Close Reading

Figure 1.1: Distant reading is often performed with aggregate visualizations of the data (as on the left), while close reading involves the careful dissection of passages down to the level of individual sentences and words (as on the right).

In the humanities, this sort of "zoomed-out" analysis is called **distant reading**. First coined by Franco Moretti (Moretti, 2005), this term is used to describe the process of understanding a corpus of texts in aggregate, generally through some form of algorithmic analysis. This was posed in contrast to **close reading**, the more common practice within the field of carefully dissecting individual passages

of text to derive in-depth interpretations. Existing visualization tools typically consider just one method or the other, and in fact the two are often considered to be at odds with one another. However, I will argue that the real value comes from visualizing them *together*: connecting them through interaction in a workflow that allows researchers to move smoothly between levels of abstraction.

In this dissertation, I will describe the importance of visualization for large-scale text analysis, in particular as related to its ability to merge the practices of distant and close reading. My specific focus will be on analysis that makes use of a particular modeling technique called **probabilistic topic modeling**. Topic modeling operates under the assumption that words appearing in the same documents are likely to be semantically related to one another—and that the more often they appear together, the stronger that relationship likely is. Topic models attempt to draw out collections of words that often appear together in the same documents, and as such may collectively represent a semantic concept. There are a variety of algorithms for doing this, but in general, topics are represented as probability distributions across the vocabulary of words and documents are represented as probability distributions across the set of topics. These vector-based representations of documents and topics allow researchers to perform tasks such as summarizing document context, computing similarities, tracking the rise and fall of topics over time, and many more.

It is my thesis that **we can direct researchers to meaningful findings within topic models by combining exploratory high-level visualization with the ability to make explanatory forays into the passages themselves**.

### Techniques and Systems

I will prove this thesis in two parts. In the first part, I will describe techniques and systems that I have developed for the visualization of topic models, both for a single model and for comparison of models. These techniques were developed during a many-year partnership with a set of humanities scholars. As part of an effort to bring data visualization and statistical analysis to bear on digitized early modern literature, I worked in close, iterative collaboration with these scholars to understand their questions and ways of thinking about their data. My primary strategy in development was that of task-driven design, focusing first on the questions that our collaborators and other users would try to answer and then laying out techniques to help address those questions.

Within a single model, a primary task I seek to support is *serendipitous exploration*: helping researchers uncover connections and patterns within the corpus other than what they might have been expecting. To this effect, I offer methods which form a workflow allowing researchers to transition smoothly between multiple levels of abstraction: the corpus level, the document level, the passage level, and the word level. Users are able to discover patterns and trends at higher, "zoomed-out" levels and then trace those patterns down to the underlying words which most influence them, seeing the words in context so as to evaluate their own hypotheses about why the high-level pattern is present. Such a workflow combines the strengths of both close and distant reading.

While exploration of a single model is the primary way through which researchers will interact with and build understanding about the documents, the parameter space of topic modeling is enormous. As such, many possible models exist, and it is important for researchers to be able to make comparisons between them so they can choose the model that best helps them answer their questions. While some limited comparison is possible by visualizing and exploring a single model at a time, greater benefit can be provided by affording comparison that is directly tuned to the uses to which the models are to be put. To this end, I have cataloged a set of topic model comparison tasks, derived from a set of single model tasks that correspond to the tasks I have observed both in my collaborations and in the literature. For each of these tasks, I lay out visual techniques for performing them, and illustrate their effectiveness in a set of use cases on domain datasets.

**Perceptual Evaluation**

In the second part of my dissertation, I will describe experiments evaluating the perceptual validity of low-level encodings that I use in my visualization of topic models. The need for such validation comes from a philosophy of visualization evaluation that combines both *holistic* and *reductionist* evaluation (Correll et al., 2014). The central tenet of **holistic evaluation** is that a system is more than just the sum of its parts. Therefore, developers must evaluate a system as a whole, as used on real data. This strategy is best equipped to answer the question "Does this system meet the needs of its users in context?" Metrics for holistic evaluation are directly related to the goals of the system's users: what insight were users able to gain from this system, and how were they able to put that insight to use? Did their insight lead to new practices or publications? Did they cure cancer or win a Nobel prize?

The problem with pure holistic evaluation is that it does not produce knowledge that is *generalizable*. The most that can often be said by a successful holistic evaluation is "This system works." Such a statement offers no new understanding to the field and does little to help those in the future who may be trying to address related tasks. In contrast, **reductionist evaluation** seeks to understand *why* a system works. To do this, practitioners break down the system into its component parts so as to be better able to perform controlled experiments upon them. In these experiments, they seek to understand how low-level design choices about the data encodings influence the way in which users perceive the underlying data.

Ultimately, both types of evaluation are necessary. As practitioners of visualization, we must validate systems in context to determine that we are providing utility to our end users, but must also ground our encodings in perceptual theory to make sure that the data users perceive matches the data being encoded. We also want to develop a greater understanding of how people perceive visualizations so that we can build better ones in the future. The first part of this dissertation offers holistic evaluations of the techniques it describes in the form of use cases developed in close collaboration with domain experts. The second part offers a lower-level, reductionist look at a set of visualizations that I make use of for conveying topics to users.

As described above, a topic within a topic model is a probability distribution across a vocabulary of words that often appear in the same documents. Common strategies for conveying these distributions include word lists (ordered by frequency within the topic), bar charts, and word clouds (with words sized correspondingly to their topic frequency). While there are some existing evaluations of these visualizations, none address their use specifically for topics. In particular, I was interested in evaluating encodings for the task of **gist-forming**: looking at a collection of words and building a cohesive idea (or "gist") of how they are all connected. Chapter 5 describes a set of experiments comparing visual encodings of topics for this task.

Chapter 6 takes a more in-depth look at the data encoding integral to word clouds: font size. Though using font size to display values associated with words (such as topic frequency) is an aesthetically pleasing and common way of showing data, there are concerns that there are characteristics of words' *shapes* that could bias perception of these values. I describe a set of experiments that identifies these biases and quantifies them, so as to inform designers of which situations need to be treated specially when using such encodings. Together, the experiments

described in this part inform both my own visualization design as well as providing generalizable knowledge for the field.

## 1.1 Contributions

In this dissertation, I will approach the visual exploration of topic models from a variety of angles. Working from an understanding of the literature and experience working in close concert with a set of domain collaborators in the humanities, I will analyze tasks for both single model exploration as well as model comparison, and offer techniques and systems to address these tasks. I will offer use cases highlighting the utility of these techniques in context with domain datasets, as well as offering low-level, perceptual evaluations of encodings being used by the systems I describe. This will lead to systems-level, design, and theoretical contributions. The main contributions of this dissertation are the following:

- Techniques and systems for exploring topic models
  - A suite of techniques for exploring single models, affording the pursuit of insight across multiple levels of abstraction.
  - An implementation of these techniques called Serendip that has been used to uncover new insight about collections of historical documents.
  - A catalog of tasks associated with exploring single topic models, and a mapping of these tasks to corresponding tasks associated with topic model comparison.
  - A suite of techniques addressing these topic model comparison tasks.
  - Use cases compiled in collaboration with domain researchers on real-world data showing the utility of these techniques in practice.
- Evaluations of visual encodings for text data
  - A comparative evaluation of visualizations for topic representation within topic models, desribing the relative benefits of word lists and word clouds for gist-forming.
  - An evaluation of font size as a data encoding, highlighting perceptual biases associated with the encoding and offering a proof-of-concept method for correcting these biases (when necessary).

# 2 BACKGROUND

*That nature which contemns its origin*
*Cannot be bordered certain in itself.*

— ALBANY, *King Lear*

This chapter will briefly introduce background information and literature relevant to the work being presented in this dissertation. The purpose of this chapter is to provide context for the problems that the ensuing chapters set out to solve. Additional related work specific to each chapter will be introduced in later chapters as appropriate.

## 2.1 Topic modeling

Topic modeling is a type of text processing that determines major themes within a collection of texts through statistical analysis. 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 for documents to be assigned to topics by considering the words they contain. Most topic modeling techniques are probabilistic, so the assignments produced are weighted.

There are a variety of algorithms and techniques that fall underneath the umbrella of "topic modeling," including things like probabilistic latent semantic indexing (PLSI) (Hofmann, 1999) and non-negative matrix factorization (NNMF) (Choo et al., 2013). The most common form of topic modeling is Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which itself has many forms, including dynamic topic models (Blei and Lafferty, 2006) and hierarchical topic models (Griffiths and Tenenbaum, 2004). LDA has gained tremendous popularity since its introduction in 2003, and is used in a wide variety of fields including studies in academic literature (Chuang et al., 2012b), social sciences (DiMaggio et al., 2013), journalism (Zhao et al., 2011), and the humanities (Jockers, 2013), just to name a few. This popularity is likely influenced by (while at the same time contributing to) the large number of open-source tools available for building LDA models, most prominently including Mallet (McCallum, 2002) and GenSim (Řehůřek and Sojka, 2010). To contain the scope of this dissertation, I will predominantly be looking at models built using LDA. However, I believe that my techniques and larger findings extend to many other models of text, which I will discuss further in Chapter 7.

LDA is a generative model that starts with the assumption that documents were created by sampling from two sets of distributions: **topic distributions**, which are distributions over the vocabulary of words; and **document distributions**, which are distributions over the set of topics. These two sets of distributions are assumed to resemble Dirichlet distributions. Given a corpus of text, these distributions can be inferred through Bayesian inference—typically using either a variational Bayes

approximation (Blei et al., 2003), or more commonly Gibbs sampling (Griffiths and Steyvers, 2004). Through this inference process, topic distributions are generated whose words tend to appear in most of the same documents within the corpus, while documents are able to be treated as weighted combinations of multiple topics.

There are a number of advantages that arise from being able to represent documents in such a way. Reducing a document's complexity down to a relatively low-dimensional vector makes it possible to discover meaningful clusters and similarities in the otherwise sparse space. Additionally, allowing the document features to be *distributions* across words helps account for the hazy, overlapping nature of natural language: e.g., given a sufficiently wide vocabulary, two documents might be discussing the same topic without using any of the same words. An LDA model can still uncover the similarity of such documents, provided that the words being used overlap enough elsewhere within the corpus.

However, there are downsides to the complexity of topics as document features, especially for domain researchers and practitioners trying to make sense of a topic model. The overlapping nature of topics can make reasoning about how specific documents fit into a trend or cluster difficult, and even understanding the meaning of a single topic can be challenging. Visual tools and representations of the model, therefore, can be instrumental to both understanding the relationships between documents (as I will discuss in Part I), as well as helping researchers understand topic contents (as I will discuss in Part II).

## 2.2 Topic model and document visualization

The nature of probabilistic topic models makes them difficult to interpret, and the need for visual tools has been identified before (Blei, 2012). In particular, the fact that the data tends to be noisy and variable makes direct interpretation difficult: indeed the strength of the models is that inferences are often built by combining many small things. Another issue is the variety of tasks involved in working with topic models, ranging from evaluating and tuning models to observing trends in topics to finding thematically similar documents.

Most tools for topic model visualization focus on specific tasks and questions by providing specialized views. For instance, Dissertation Browser (Chuang et al., 2012b) uses models built on PhD dissertations to track inter-department collaboration. Other techniques are primarily concerned with tracking topic evolution

through time, including (Cui et al., 2011; Havre et al., 2000; Wei et al., 2010) which use "river flow" layouts. I make use of such layouts for making comparisons over time as described in Chapter 4. Termite (Chuang et al., 2012a) is a tool for understanding topic models *themselves*, rather than using them as *tool* for exploring the corpora. I draw inspiration from Termite for some of my encodings described in Chapter 3, though I am ultimately more concerned in that chapter with exploring relationships between documents rather than those between words.

A broad range of work has considered using visualization to explore text corpora beyond topic models. A survey of some of the techniques for doing so within the humanities can be found by Jänicke et al. (2015). A common strategy is to abstract the texts as glyphs and position them in 2D as a scatterplot. Numerous approaches for organizing these layouts exist (Endert et al., 2011; Joia et al., 2011; Paulovich and Minghim, 2006), with other work focusing on user control (Endert et al., 2012b) and understandability (Brown et al., 2012; Gleicher, 2013). I apply the idea of flexible layout as a mechanism for using topic model data.

Existing tools have identified *individual* documents as an important unit of study as well—though rarely the same tools that visualize documents at the corpus level. Within most topic model visualization tools, single documents are either inaccessible or viewable only as plain-text. This is generally sufficient, as model corpora typically contain documents on the order of abstracts, which can be easily skimmed. When modeling much larger documents like books, additional information from the model is needed to direct the user through the document's structure. Others have employed tagged text displays for such overlay of information (Correll and Gleicher, 2012; Correll et al., 2011), and they have been shown to allow users to make aggregate judgments without sacrificing readability (Correll et al., 2013). Plaisant et. al. have used colored tags to indicate metadata and user interest (Clement et al., 2009; Plaisant et al., 2006). I have built on the use of such tags in an attempt to interactively convey the probabilistic uncertainty of topic models, reflecting not only which words belong to which topics, but which words are *important* to those topics. This is discussed further in Chapter 3.

I believe that one of the most important aspects of my approach to topic model exploration is the ability to perform analysis that flows smoothly across levels of abstraction, from the aggregate to the specific. Before beginning my work, Jigsaw (Stasko et al., 2008) was one of few text visualization tools to consider multiple levels, offering zoomed-out visualizations of the corpus while providing plain text views of the documents within accompanying statistics. PaperLens (Lee et al.,

2005) similarly combines multiple views of clean metadata. However, neither of these tools considers topic modeling at the heart of their analysis, nor do they afford the same corpus-document-passage-word workflow that allows users of my techniques to understand topic trends not just within short abstracts but within longer documents. Since the publication of some of the work presented in this dissertation, other techniques for such multi-tiered exploration have arisen, including VarifocalReader (Koch et al., 2014). However, there is still much to do to fully integrate the processes of close and distant reading in large-scale text analysis.

## 2.3 Perceptual evaluation of encodings

As discussed in Chapter 1, for evaluation of visualization tools and techniques, it is important that top-down, holistic evaluation be paired with bottom-up, reductionist evaluation. Holistic evaluation lets us confirm that our systems work for users in practice, while reductionist evaluation helps us understand *why* they work, offering us knowledge that can lead to generalizable theory. While Part I of this dissertation focuses more on systems and techniques, along with their accompanying top-down evaluation in the form of use cases, Part II focuses on the bottom-up, perceptual evaluation of the encodings used in them. Many of the encodings I employ have been validated by previous literature, as I tried to make use of existing viable techniques when possible.

#### Reorderable matrices

The primary corpus-level visualization that I employ in Chapter 3 is a **reorderable matrix**. Reorderable matrices have been used since they had to be rearranged by hand, as seen in Jacques Bertin's work from 1967, reprinted in Bertin (2011). Often, the goal for designers using reorderable matrices is to algorithmically pick the ordering that will best expose patterns of interest (Climer and Zhang, 2006; Henry and Fekete, 2007). I make use of such "optimal orderings" in the comparison heatmaps described in Chapter 4. However, it has also been shown that people can make use of reorderable matrices if they are given direct control of the orders themselves. Siirtola et al. ran studies in which participants from a variety of fields were able to find interesting attributes and patterns within the data in their first time interacting with reorderable matrices (Siirtola, 1999).

This work encouraged us to give users the interactions to find their *own* optimal orderings when exploring topic models, as described further in Chapter 3.

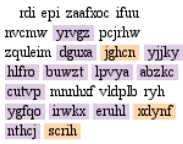
### Tagged text

At the level of augmenting researchers' close reading practices, the most important encoding that I use is **tagged text**. This is a method of encoding information pertaining to specific words in which individual words are marked (generally with a colored background) to indicate their associated properties or data values, and has been used in a variety of visualization systems (Clement et al., 2009; Correll et al., 2011; Plaisant et al., 2006). Tagged text can help inform the reader as to what textual details contribute to the overall pattern and can help them localize patterns in the larger text. However, for such tagged text visualizations to be useful, the reader must still be able to infer the larger trends from lower level details in specific words. If we are to use them as a method of connecting close to distant reading, it is important to confirm that tagged passages accurately reflect the degree to which they are representative of a whole.

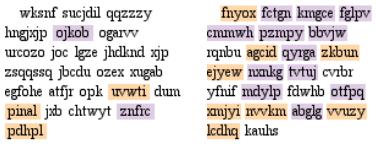
In a paper written with Michael Correll and Michael Gleicher, we tested subjects' ability to make accurate judgments about the *proportion* of colored tags within a passage of tagged text (Correll et al., 2013). If we were going to encourage researchers to make inferences about a passage based on the perception that it contained, for example, roughly 70% of one category and 30% of another, we wanted to make sure that these perceptions were accurate. In a series of five experiments, we tested participant performance on a variety of such estimation tasks across a number of factors. We came away with the following results (summarized also in Figure 2.1):

- Participants were able to make accurate (within a few percent) and efficient (faster than counting) judgments about relative tag proportions. These judgments were robust to tag density and uniform variation in word length.
- Certain combinations of colors (e.g., red and green) can skew these judgments and ought to be avoided in design.
- Systematic differences in area between tags can also skew this perception. However, artificial adjustments to the tags can mitigate this bias.

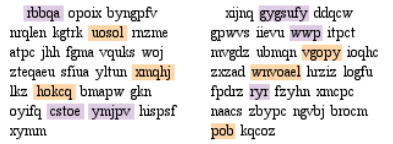
These takeaways provided both validation for my use of tagged text encodings for such tasks as well as design guidelines on how best to ensure good participant performance when doing so.



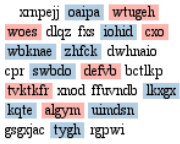
(a) Viewers accurately estimate the proportion of tags (e.g., if the text is 20% or 60% orange) to within ~5%



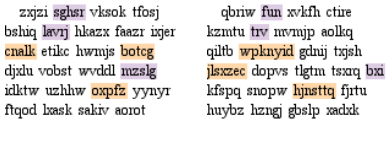
(b) Estimation accuracy is robust to changes in density (e.g., whether 20% or 80% of all words are tagged).



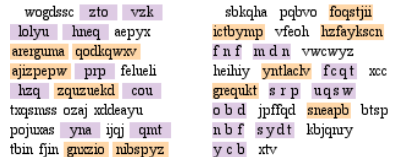
(c) Estimation accuracy is robust to variance in word length (e.g., if words can be 5, 4-6, or 3-7 letters long).



(d) Color can bias tag estimation. Some colors may systematically be selected more than others (e.g., commonly selected red versus commonly avoided green).



(e) Tag area can bias estimation of tag number. Larger areas may seem to have higher counts. E.g., on the right, orange words are longer than purple words.



(f) Area bias can be mitigated by adjusting the padding (i.e., extra space around words) or tracking (i.e., inter-word spacing) of tagged words.

Figure 2.1: A summary of the results of the five experiments reported in our paper evaluating tagged text encodings (Correll et al., 2013). Together, they suggest that tagged text displays can be a useful presentation of data that accurately conveys the overall proportions of tags while allowing the reader to see the individual words, providing some design guidelines are followed.

### Topic representations and word size encodings

The final low-level encoding I am interested in is how best to convey topic distributions to researchers. There are a variety of different methods that are employed in the literature. By far the most common is the use of **word lists** (Figure 2.2a): simple lists of words contained within the distribution, generally ordered by frequency (Blei et al., 2003; Chaney and Blei, 2012), but occasionally ordered by other metrics (Chuang et al., 2012a). **Word clouds** are an additional method that are increasingly being used in topic visualizations (Cui et al., 2011; Meeks, 2012; Wei et al., 2010). In these visualizations, each word's font size is assigned proportionally to its probability within the topic distribution (Figure 2.2b). Bar charts are


(a) Word lists generally display a topic's words in descending order of frequency.

(b) In word clouds, words are positioned along a spiral out from the center, sized proportionally to their topic frequency.

(c) Bar charts can be used similarly to word lists, but offer additional information about the relative frequency of the words.

Figure 2.2: The most commonly seen topic representations in the literature are word lists, word clouds, and bar charts.

occasionally used, as well (Alexander et al., 2014; Torget et al., 2011), providing similar information to word lists, but with additional indication of proportions within the topic (Figure 2.2c).

There are two main questions I seek to answer in respect to these encodings. The first is to what degree people are able to *comprehend* topics being represented in different ways. Success at this task is harder to measure than for the encodings described earlier in this chapter. With reorderable matrices, we want users to be able to find important patterns and similarities when present. With tagged text, we want users to be able to make accurate judgments about tag proportion. For our topic encoding, we want users to be able to form an accurate judgment about how the words being presented are *semantically* related (or not). Due to the broad interpretations of semantic meaning, this is a task for which it is difficult to determine a ground truth. To be able to evaluate these interpretations, I defined the high-level task of **gist-forming** and created a corresponding experimental task to measure it. These are discussed in Chapter 5.

The second question I seek to answer is to what degree people looking at these topic representations can accurately perceive the proportions being encoded. This question is for the most part specific to word clouds: word lists do not offer anything more than order, and bar charts have been more thoroughly evaluated by others (Cleveland and McGill, 1984). However, word clouds (and visualizations that employ font size as a data encoding more generally) are rather contested within the literature. In one study questioning the use of word clouds to convey data, Rivadeneira et al. studied participants' performance using word clouds on

a variety of different tasks (Rivadeneira et al., 2007). Though they found that word clouds serve users poorly for many of these tasks—most especially search, supported also by Halvey et al. (Halvey and Keane, 2007)—their use cases differ dramatically from that of gist-forming (discussed further in Chapter 5).

Other critiques, while not adding experimentation, describe the perceived strengths and weaknesses of word clouds for other tasks. Viégas and Wattenberg assert that given word clouds' broad appeal, there must be something worthwhile about them as an encoding (Viégas and Wattenberg, 2008). Word clouds are admonished in a well circulated blog post titled "Word Clouds Considered Harmful," but the criticism is more about poor journalistic practice than a commentary on gist-forming capabilities (Harris, 2011). Hearst and Rosner point out that the visual appeal and social nature of tag clouds give them inherent value, possibly separate from quantitative task performance (Hearst and Rosner, 2008). Finally, Meeks discusses the use of word clouds with topic models, citing in particular their compact representation (Meeks, 2012). However, he does not seek to make his justification empirically.

While much of the debate surrounding word size encodings pertains to word clouds specifically, these encodings are used in a variety of other applications, including cartographic labeling (Afzal et al., 2012; Skupin, 2004) and a number of different hierarchical visualization tools (Brath and Banissi, 2015; Wattenberg and Viégas, 2008). In Chapter 6, I describe a set of experiments designed to measure the accuracy of comparative judgments made within visualizations using these encodings in an attempt to better constrain their use in visualizations broadly.

# Part I

# Techniques and Systems for Topic Model Exploration

# 3 EXPLORING TOPIC MODELS

*Alas, poor shepherd, searching of thy wound,*
*I have by hard adventure found mine own.*

— ROSALIND, *As You Like It*

Exploration and discovery in large text corpora can be a daunting task. Corpora can easily grow to thousands or more texts, ranging in length from short snippets to long books. The task is further complicated by the range of questions that can be asked of such corpora, broad both in subject (making comparisons across time, genre, author, etc.) and in level of detail (corpus, document, passage, even word). Discoveries must often connect multiple subjects and levels of inquiry. Fortunately, there is considerable information to aid these inquiries. Beyond the texts themselves, there are statistical summaries of content, document metadata, and analysts' explicit and implicit knowledge of the documents and their context. However, mixing these different types of information across scales of inquiry is challenging. The information types, and the existing tools that support their use, generally focus solely on a particular scale.

In this chapter, I introduce a tool for text exploration with topic modeling that is designed to afford inquiry that traverses across multiple scales and merges information types. The core idea is that to enable proper hypothesis formation and understanding, a system must provide views showing the data from multiple vantage points as well as connections between different types of information, allowing readers to transition easily across scales, data types, and research questions. To achieve this, we have adapted existing visual encodings to work with text corpora data, developed new encodings that address some unmet needs, and introduced statistical methods that help connect between different object types. The resulting system enables users to explore questions about collections of texts, passages within texts, and sets of words that define topics, moving smoothly between distant and close reading along their path of inquiry. We have embodied our approach in a system called **Serendip**.

The motivations for Serendip evolved while working with literature scholars to enable the use of topic modeling to study large historical text corpora. These readers have different emphases in their use of text analysis (Correll and Gleicher, 2012), though we believe our approach can be applied broadly. Over the course of a multi-year collaboration, we have come to understand their needs for text analysis, along the way evolving a prototype to address these needs. As part of this collaboration, I spent time at the Folger Shakespeare Library—an international center for literary and historical research—working with collaborators to understand how the prototype might be adapted to address their research questions. The work identified four key goals in a text corpora exploration system.

First, the system must address **issues of scale**. One of the main strengths of

topic modeling is the way in which it enables working with corpora that are too big to read. There are two primary issues of scale to consider: large numbers of documents, and large numbers of words *within* the documents (e.g., books and plays). While scaling to many documents is commonly considered in corpus exploration systems, *long* documents present an uncommon challenge. A system must both present aggregate trends *across* documents, but also guide readers to where these trends are reflected *within* them.

Second, a system must allow for **inquiries across different scales**. Some inquiry flows from the top down: identifying trends across sets of documents may lead to the identification of *specific* documents and passages supporting them. The ability to find such support is critical for literary scholarship, where arguments are frequently won or lost on the basis of close analysis of exemplary passages rather than the distribution of figures and charts. It is also valuable in other domains, where readers must check to see if statistical patterns accurately reflect the meaning of the passages (Correll and Gleicher, 2012). Alternatively, inquiry can flow from the bottom up, starting with a word or topic of interest and seeing how these things make up larger patterns across documents. Interaction between levels is important: a question about one scale inevitably leads to questions and answers at other scales. A system must provide clear starting points for exploration, as well as ways of using intermediate results to build to next steps at potentially different levels.

Third, a system must allow readers to pull together **multiple sources of information**, both statistical as well as human-curated metadata. Some of the pieces of metadata are explicitly defined (e.g., the date of a book), while others rely on the implicit knowledge of the reader. Inquiries often mix these different types of information.

Finally, we had the goal of promoting **serendipitous discovery**: the act of finding something unexpected while looking for something else (or nothing in particular). While serendipity is often thought of as instances of individual luck, statistical chance, or divine predestination, research points to practical ways in which it can be consciously fostered (Thudt et al., 2012). This sort of design can be seen in situations as commonplace as where to put books on the shelves of a library, but is relatively unexplored in visualization.

I have built a system called Serendip that addresses these goals, providing three tightly coupled main views (see Figure 3.1). It extends a reorderable matrix view with novel features that allow it to better address issues of scale, as well as to


Figure 3.1: The three main views of Serendip: CorpusViewer, TextViewer, and RankViewer.

allow for multi-scale and multi-information fusion. It uses a tagged text encoding along with an accompanying overview visualization, adding features that direct users to key passages and convey the probabilistic nature of topic tags, improving the quality of user interpretations as well as their understanding of and trust in the model. It provides a novel view of topic words designed to address specific questions that arise and connecting these inquiries to other scales. All of these are combined with interaction techniques that allow readers to follow branching paths of inquiry across multiple scales and units of analysis. To illustrate the utility of these views and interactions, I provide example use cases of Serendip, both on a set of literature on data visualization as well as a set of historical documents with explorations driven by a literature scholar.

Work from this chapter was originally published as Alexander et al. (2014).

## 3.1 Related work

There is much from the field of text corpus visualization—particularly topic model visualization—that influenced our techniques and design. Most background and related works relevant to topic modeling and visual corpus exploration were discussed in Chapter 2. In this section, I will discuss related work specifically pertaining to *serendipitous exploration*.

### 3.1.1 Fostering serendipity

The word "serendipity" was coined by Horace Walpole, referring to a story called "The Three Princes of Serendip," in which the protagonists are able, through chance observation, to uncover the nature of a camel they have never seen. The word has come to be associated with occurrences of happy accidents that lead to unexpected discovery. Though such instances are often attributed to fate or providence on one hand or extreme cleverness on the other, research has been done to determine how to make such "accidents" more likely. The most often used example is that of looking for a book in a library: a patron navigates the stacks to find a particular book, but because of the way books are organized, they end up stumbling upon an even better book sitting on the same shelf. Thudt et al. provide a thorough survey of the research on promoting serendipity, distilling it into a set of principles that apply to the design of visualizations (Thudt et al., 2012):

1. **Providing multiple access points.** Unlike physical books on a shelf, electronic documents can be arranged in many ways simultaneously. Users can make a more diverse set of findings if they can view the data from a variety of different angles.

2. **Highlighting adjacencies.** Serendipitous findings tend to occur *near* where the user is searching, and so it is important to visually emphasize these proximities.

3. **Offering flexible pathways for exploration.** While many systems offer data access through directed querying, encouraging open-ended exploration—with a variety of viewpoints and transitions between them—seems to enhance serendipity.

4. **Enticing curiosity and playfulness.** Finally, even when presented with surprising juxtapositions, the user must be in a creative state of mind to be able to make connections between them. An experience that engages their sense of fun has been shown to promote this state of play and exploration.

Thudt's Bohemian Bookshelf system, based on these principles, helps users find books in a library. We have aimed to use the principles in our approach for topic model exploration.

## 3.2 Exploring text corpora with Serendip

Our approach is designed to combine tenets of serendipity and multi-level exploration, dealing comprehensively with issues of scale. We incorporate three main views, each designed to serve as an access point to the data and support a different level of inquiry. At the **corpus level**, we provide a reorderable matrix to highlight adjacencies between documents and topics. At the **document level**, we use overview displays which direct readers to tagged text at the **passage level**. Finally, at the level of **individual words**—a level we only observed the need for after watching users interact with our text level tool—we introduce a ranking visualization that shows how words are distributed across the topics.

Ultimately, it is the interactions *between* these levels that provide "flexible pathways for exploration" as laid out in the above principles of serendipity. There are many possible units of interest within the corpus: topics, documents, metadata, passages, words. Users may find themselves entertaining one (or some) of any of these, either as their initial entry-point to exploration or as an intermediate step along the way. To provide for flexible information usage, we offer techniques for using any of these units to identify other units of interest, of potentially different types. For example, documents (or sets of documents) can be used to find other documents, metadata categories, topics, passages, or words. Providing linkages between the different combinations of units requires an array of different visual, interaction, and statistical techniques. User inquiry must be allowed to move across these levels in a flexible, sustained way; our users need access to multiple starting points and control over their own successive re-orientations, building inquisitive momentum as intermediate results drive next steps. These linkages are described in the following sections.

The centerpiece of the corpus level tool, CorpusViewer (Section 3.3), is a re-orderable matrix that connects documents to topics. To address the "many documents" and "many topics" issues of scale, the matrix supports filtering and selection, aggregation, and ordering. Ordering is a key tool as it not only addresses scale—by placing salient objects at the top—but also promotes serendipity by placing similar objects next to each other. CorpusViewer focuses on exposing patterns across documents and topics, and identifying specific items to explore more closely. Additionally, it provides ways to overlay other information types, such as metadata and words, and links to other views.

TextViewer (Section 3.5) allows for detailed examination of how topics are

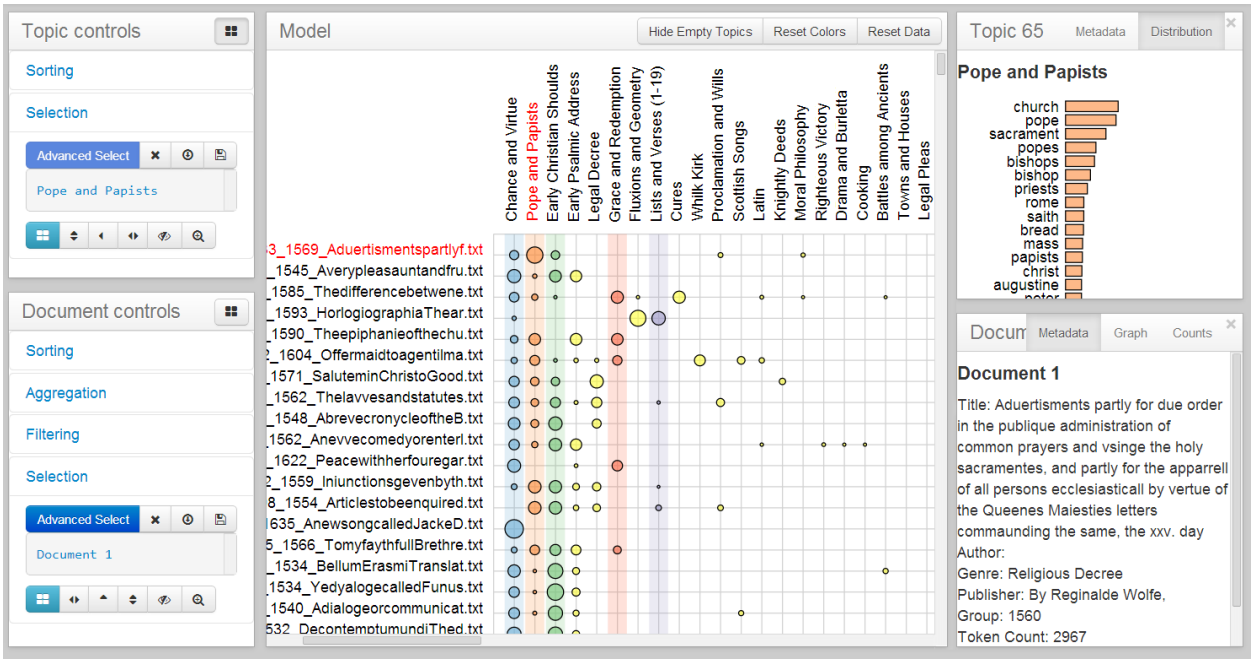



Figure 3.2: CorpusViewer centers around a re-orderable matrix that provides a variety of ordering, selection, aggregation and annotation features to help users find high-level patterns in the corpus and connect to specific documents and topics. Each row represents a document, each column a topic, and the circle size encodes the proportion. Here, colorings are applied to selected columns in order to connect to other views of topics. In the upper right, a topic is depicted by showing the proportions of its most salient words.

reflected within a specific document. A tagged text visualization shows the topics and the text. To support long documents, a summary graph shows how the topics occur over the length of the document. This view's main role is to connect high-level trends to specific example passages, validate them for the user, and help build the user's understanding in the workings of the model. However, it is also important for identifying topics and words to explore in other views.

RankViewer (Section 3.6) allows users to examine specific words and see which topics use them. This tool is useful for relating topics and words. It can provide topics (and orderings of topics) to explore more closely in other views. Central to viewing words in topics (in both RankViewer, but also in TextViewer and CorpusViewer) is a mechanism for ranking words (Section 3.4).

## 3.3 Viewing the corpus

CorpusViewer provides a high-level overview of the entire corpus, affording distant reading as described in Chapter 1. It is designed to help identify trends in documents and topics, and to use them to focus on more specific items (or sets of items). Its main view is a reorderable matrix that plots documents (rows) against topics (columns), encoding the values of the distributions as circular glyphs on the vertices of the grid (Figure 3.2). We have supplemented this matrix with features to combat scale, connect outside information, and promote serendipitous discovery at the corpus level.

### 3.3.1 Filtering and selection of data

A simple but important way to combat the potentially vast dimensions of this matrix is to make sure users are able to focus on objects of interest. We provide a query-system that allows users to pick out documents and topics based on their metadata. Once selected, these sets can be hand-tuned, colored, moved to a more prominent position in the matrix (typically the top-left corner), used as a basis for reordering the matrix as described in Section 3.3.2, or saved to be explored later.

Selections (and set building) can also be done manually. This is sometimes useful for removing erroneous rows and columns from a query. More importantly, the ability to build sets provides a way for the user to express knowledge (e.g., of a known set of objects of interest), and to use intermediate results of prior steps to make new steps (e.g., using the top elements of a sorting, or surprising anomalies, to create a new ordering). First class selection sets are supported for both documents (rows) and topics (columns).

### 3.3.2 Reordering the data

Reorderable matrices have been around since they had to be rearranged by hand (Bertin, 2011). While many try to find the “optimal” order of rows and columns in a matrix (Climer and Zhang, 2006; Henry and Fekete, 2007), people have been shown to find interesting attributes and patterns within the data if they are given direct control of the orders themselves (Siirtola, 1999). Embracing this idea, we have created a number of ordering options designed to address the requirements of our tool.

Good orderings combat scale, collecting the most salient items at the top or bottom of the list. They also promote serendipity by putting similar objects next to each other and providing different ways of looking at the objects. We identify three different types of orderings: **blind orderings** that just use the distributions, but are useful for starting an inquiry; **question-based orderings** that use other sources of information (such as sets of other units or metadata); and **similarity-based orderings** that order based on similarity to a seed set of like units. These orderings can serve as starting points of investigation (with or without a specific question or object in mind), or serve to use an intermediate result or finding as a point for further inquiry. In Serendip, we have provided orderings of each type for both the rows (documents) and columns (topics) of the matrix.

**Document orderings**

The number of documents (and therefore rows) contained within a corpus is perhaps the biggest scaling issue in these models, and therefore good techniques for document ordering are crucial.

**Blind:** As a blind entry-point into the documents, we offer a way of sorting by their topical complexity. This was inspired by (Dou et al., 2011), which used a form of entropy to distinguish between documents that contain single topics and those that contain multiple topics. Rather than using entropy, we allow the user to sort documents by the proportion of their $n^{th}$ strongest topic. This can provide an approximation of topic entropy. For example, sorting by the third highest topic proportion finds documents with (at least) three strong topics. Sorting by small numbers of topics (especially 1) finds documents that are strongly dominated by a single topic.

**Question-based:** Question-based orderings attempt to answer user queries by pulling the most relevant documents to the top of the list, where relevancy can be based on information other than the documents. The most commonly used is ordering based on the strength of a selected topic (or set of topics) of interest (e.g. "Which documents are highest in topic $x$?"). Documents can also be sorted by metadata fields, which can be useful for exposing topic trends in a particular group of documents or for aiding in search. (Some of these same tasks can also be achieved by using metadata for searching, filtering (Section 3.3.1), and aggregation (Section 3.3.3).)

**Similarity:** Creating orderings of documents by similarity of their topic data—e.g. finding documents that resemble an example document or set of examples—is

a particularly important ordering tool. It provides an initial entry point when the user has familiarity with even a few of the documents, but also provides a follow-up step when interesting documents are found. There are two key questions in building similarity metrics: how to compare document vectors, and how to calculate distance to a group.

The documents are represented as vectors of their topic proportions. In general, meaningful distance metrics in high-dimensional spaces are challenging (Zimek et al., 2012). Document vectors are also sparse and (nearly) convex (discounting truncation issues). We generally use the cosine similarity metric rather than Euclidean distance, but both metrics are mainly meaningful when documents are close: as documents become farther apart, the metrics become less interesting. We use Euclidean metrics for performing clustering. Our system also provides *weighted* variants of similarity metrics, where different topics are weighted differently. The weights can come from various forms of topic selection and ordering criteria (below), providing a simple form of distance metric adaptation.

The primary way that our system measures distance to a set of documents is by computing the "center" of the set (by computing the mean), and measuring the distance to this point. This approach has two key flaws: first, averaging vectors damages the sparse and convex structure; second, the center may not adequately capture a multi-modal (or oddly shaped) distribution. To combat these issues, we provide an alternative. We generate multiple centers for the set by k-means clustering, and define distance as the minimum to any one of the centers. As a special case, choosing the number of clusters equal to the set size guarantees that items in the set have zero distance to it. The multi-center approach, in principle, improves performance because the distances are smaller and therefore are better approximations. In practice, the errors in the single center approach often create serendipitous accidents: while the closest documents to a set are often not in the set, they are nonetheless often interesting and/or surprising. When computing distance to a center, we can use the variance of the set to provide a weighting so that higher variance topics contribute less.

**Topic orderings**

While there are generally considerably fewer topics than documents, it is still often impractical to scan through the entire list. Column orderings are also important for promoting serendipity: not only for identifying other topics of interest, but for seeing that documents combine multiple topics in different ways. Column

orderings can also be useful for suggesting smaller sets of relevant topics so as to have a more focused distance metric for sorting rows.

**Blind:** For the topics, we offer a variety of blind entry points based on statistical metadata. Most prominently used among these is sorting the topics by the number of documents containing them, giving the user a sense of the most prevalent topics in the model. Another valuable ordering is the variance of a topic's proportions when it is present within a document: does it tend to dominate documents, or is it more briefly mentioned? Other blind orderings include maximum proportion, minimum proportion, and mean value.

**Question-based:** The most common method for ordering topics is based on a document (or set of documents). The topics are sorted by proportion in the document (or average in the set).

A second set of question-based orderings pulls in outside information in the form of document metadata. These orderings use statistical measures to determine how well different topics correlate with metadata distinctions. The more general of these orderings is the ANOVA ordering. This tool uses a categorical metadata element (such as genre) and performs a one-way ANOVA for each topic to determine how likely the different categories are to have different mean values. Sorting by the F-value ranks the topics by their ability to distinguish the different categories. A second such ordering tool is contrast ordering. This tool takes two sets of documents and computes the t-statistic for each topic, testing whether the two sets have significantly different mean values. Again, ranking by this statistic orders the topics by how well they distinguish between the two classes.

We note that topic data does not meet the assumptions for the statistical tests applied to produce orderings. However, since our goal is to assess the relative values for ranking, rather than using the precise values to determine significance, we feel these approximations are justified. Alternate statistics, such as the non-parametric Kruskal-Wallis H, could be applied instead (see Section 3.9).

**Similarity:** Much like documents, topics can be sorted by their similarity or weighted similarity to a particular topic or set of topics. The metrics compare how the topics are used in the documents, rather than the words they contain. By default, cosine distance is used to compare vectors; however, an alternative uses the Spearman rank correlation coefficient to measure how similarly the topics rank the documents. In practice, these seem to provide similar results.

**Other Proximity Displays**

Ranking, often using distance metrics, is an important method for creating serendipity by putting similar things next to each other in a list. However, the confines of a 1D ranking may not adequately capture nearness, and other visual encodings of similarity may promote serendipity in different ways. For these reasons, our system also generates scatterplots of two dimensional embeddings of the distance functions. The default is to use a spectral embedding as it captures the near-neighbor behaviors that are most likely to be interesting, and ignores larger distances that are less likely to be meaningful. Non-linear manifold embeddings, such as IsoMap, have similar properties. Our implementation uses a standard library called scikit.learn (Pedregosa et al., 2011) that provides a number of embedding techniques. Scatterplots are colored by metadata, as shown in Figure 3.8.

To create a very different view of proximity, our system performs a k-means clustering and presents the results with each cluster being a list ordered by distance to the cluster center. This view emphasizes neighborhoods of similar objects to promote the serendipitous discovery. It also provides a sense of the diversity in the corpus, as the cluster centers provide a sampling of dissimilar documents.

### 3.3.3 Aggregating the data

While a variety of ordering metrics combats scale by concentrating important documents together, this is not always enough, especially when trying to compare *groups* of documents. This connects to our goal of pulling outside information into the analysis: comparing collections of documents—especially those grouped by categorical metadata like genre or conference, as seen in Section 3.8—is a very common use case. We enable such comparison by allowing the user to aggregate documents into sets based on arbitrary fields of metadata. This can dramatically reduce the size of the matrix to be explored.

When aggregating, we average document rows into single vectors that display the mean value of each topic's proportion using filled circular glyphs. Our encoding also reflects the *variance* of topic proportions within these groups of documents. On top of the filled circles, we add three thinner, unfilled circles that encode the first, second, and third quartiles of the aggregated values (see Figure 3.3). In other words, if a set of documents varies dramatically in its proportion of a particular

topic, that glyph will resemble a bulls-eye of concentric circles. If the documents all share similar proportions of the topic, the concentric circles will fall roughly on top of one another, approaching a glyph that looks like a single circle.

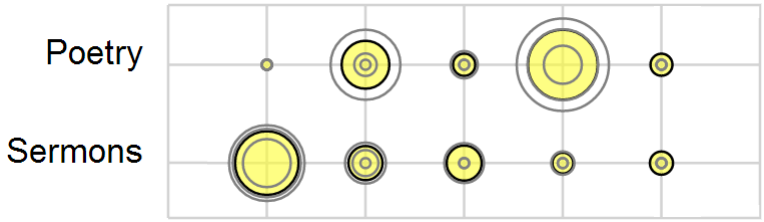



Figure 3.3: When aggregating documents, CorpusViewer uses glyphs that give a sense of both mean and variance. The filled circle represents the average proportion for a given topic for all documents in the aggregation. The three non-filled circles indicate the first, second, and third quartiles for this topic's proportions across the documents in the aggregation.

### 3.3.4 Other features for exploration

There are a number of features for annotation within the tool. The user can label documents with any field of metadata; title is a common choice. While there is no such associated metadata for topics, the reader can rename topics with arbitrary strings of their choosing (see Figure 3.2), creating meaningful labels based on their observations across words and documents. By combining multiple sources of information, users can come up with names that are much more descriptive and interesting than what can be generated algorithmically (see Section 3.8.2). Readers can also assign colors to sets of documents and topics (see Figure 3.2), either manually or by queried selection. These colors are retained across all levels of Serendip (see Sections 3.5 and 3.6).

CorpusViewer also provides extra details on demand in the two windows on the right that give statistical information and metadata about selected topics (top) and documents (bottom) (see Figure 3.2). The topic view is particularly useful for viewing the topic's highest ranked words, as described in Section 3.4. Finally, these windows act as jumping off points into the other levels of Serendip. Double-clicking the document's heading will open the document within a new TextViewer window (Section 3.5) while clicking on a particular word in the topic view will display that word's rankings within a new RankViewer window (Section 3.6).

## 3.4 Viewing topics

As discussed in Chapter 2, one of the most important encodings within a topic model visualization is that which conveys a single topic. This requires deciding upon a method for displaying the top words within a topic, along with a metric for *ranking* the words. This section describes the topic representations used in each view within Serendip.

#### Ranking words within topics

As described in Chapter 2, topic distributions cover the entire corpus vocabulary (at least in theory). As such, any visualization of the distribution can only display a subset of these words. Representing a topic's words therefore requires a good metric for ranking them. In addition to providing a ranking to the topic visualizations, this metric determines color-ramping in TextViewer (Section 3.5).

The most commonly used ranking metric is frequency: the percentage of a given topic accounted for by each word. This has a distinct bias toward words appearing in *many* topics. Models typically factor out pervasive stop words such as articles and pronouns (a process which is discussed further in Chapter 4), but the most frequent words in a topic are often still pervasive enough to be somewhat uninformative for *distinguishing* topics.

The other extreme is to rank words within a topic by the information they gain toward identifying that topic. Information gain can be computed by using the Kullback-Leibler divergence (Kullback and Leibler, 1951) between the probability an arbitrary word $k$ was generated by topic $T$ *given* $k = w$, and the probability $k$ was generated by $T$ *without* that knowledge. Intuitively, ranking words by information gain in this way will pick out words that best distinguish the topics. However, this metric has a large bias toward very rare words that appear only a handful of times within the corpus, and are therefore uninformative in their own way.

We create a metric that combines the benefits of both frequency and information gain by multiplying them together. This *saliency* metric is similar to that introduced by Chuang et. al. for finding salient words across an entire model, not just within a topic (Chuang et al., 2012a). Within Serendip, we allow users to rank by any of these metrics.

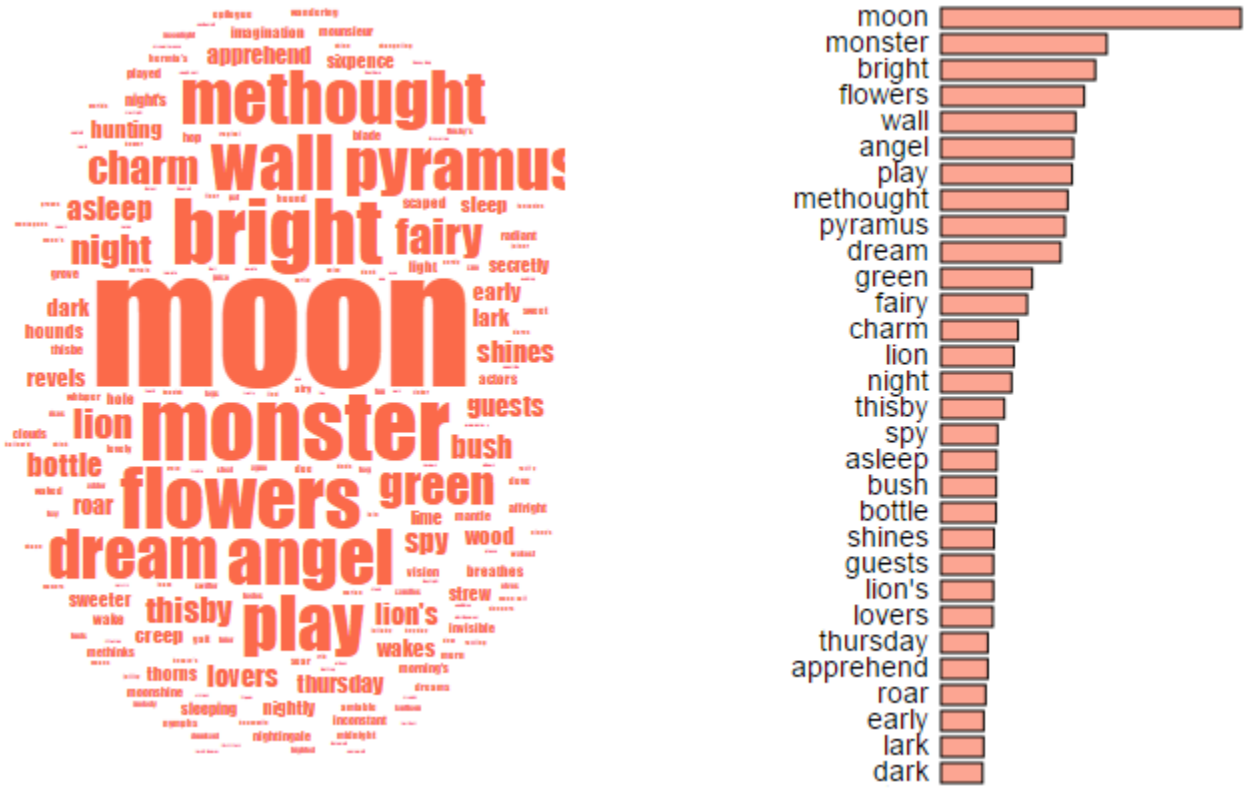



Figure 3.4: For topic representations within CorpusViewer, TextViewer, and RankViewer, we allow researchers to decide between bar charts and word clouds to display the top-ranked words.

**Displaying topic rankings**

Serendip allows topic distributions to be displayed as either bar charts or word clouds, per the choice of the researcher (Figure 3.4). Bar chart displays are ordered by the metric of the researchers choice (frequency, information gain, or saliency) in descending order, with bars indicating the relative proportions of the metric for each word. Word cloud displays are laid out in a spiral formation, with the highest-ranking word in the center and lower ranking words being placed further and further towards the edge of the display. Font size of these words are proportional to the word's ranking metric. In initial development, we had initially only provided support for bar chart topic visualizations, but implemented word cloud displays after determining that they were well-suited to our tasks (see Chapters 5 and 6). Interactions with collaborators seem to indicate that these (slightly more aesthetic) visualizations can be more engaging. One collaborator particularly enamored with the word clouds dubbed them "topic fingerprints."

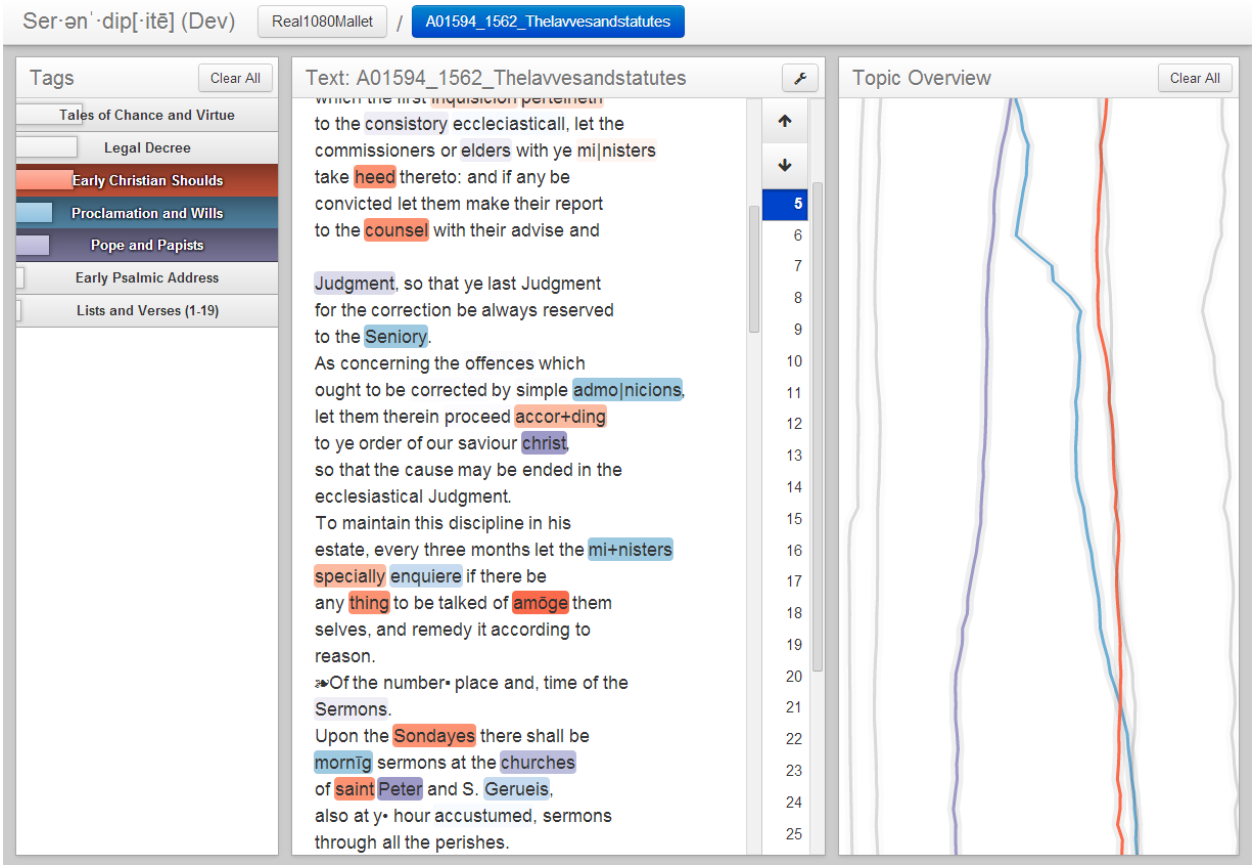



Figure 3.5: TextViewer combines tagged text with a line graph overview for navigation. The line graph can be used to navigate to passages with varying densities of topics. The tagged text is ramped so that words with higher ranking (see §3.4) are darker and more salient. Topics can be toggled on and off with buttons to the left.

## 3.5 Viewing documents

TextViewer not only allows the viewer to see specific documents, but also to see how various topics are reflected within them. This lets well known passages serve as entry points to the model (by suggesting topics for exploration at a higher level) as well as being a way of moving from distant to close reading, providing exemplary passages for high level trends. The need to trace trends down to the passage level is particularly prevalent among humanities scholars, for whom textual examples are a required part of their rhetoric. However, providing low-level examples can also help readers in other domains, both to explain high-level trends and to build trust in the model (Correll and Gleicher, 2012).

### 3.5.1 Intra-document navigation

Topic model visualizations that *do* give access to the documents typically present the raw text, unannotated. This is sometimes sufficient when the documents being modeled are on the scale of abstracts. If the model assigns a particular topic to a given document, abstracts can be quickly skimmed as a sanity check. Such is not necessarily the case when modeling documents on the order of novels and books. Just as themes and subject matter will come and go throughout the course of a story, so do the occurrences of a topic vary in density. As such, readers may require a navigational aid to find exemplary passages of high-level topical trends within longer documents. We use overview visualizations to direct readers in this manner.

A variety of existing techniques for representing document structure were more complicated than necessary for the task (Collins et al., 2009a; Keim and Oelke, 2007; Rohrer et al., 1998). Our overview is simple: a line graph displaying densities for each topic with adjustable smoothing. These graphs can show many topics at once, with users being able to toggle topics on and off, giving "on" topics a qualitative color encoding—pulled from (Brewer et al., 2003)—that is consistent across the different levels of Serendip. This makes comparisons of topic trends perceptually clear and affords smooth transition of the researcher's exploration across levels.

In TextViewer, these graphs become a useful aid to navigation within a document. It is easy to determine from the peaks and valleys of the topic lines which passages are high or low in a topic, which contain a *mixture* of topics, etc; by clicking a particular position on the overview, the reader is able to easily scroll to any passage of potential interest within the pages of the document.

### 3.5.2 Text tagging

Once the reader has arrived at a passage, their question becomes: which words matter? Providing the raw text offers some utility, but we can provide more by annotating the text with data from the model. In TextViewer, we do this using colored backgrounds to highlight individual words. Since LDA labels each word with a topic, the easiest approach would be to just assign each topic a single color. However, tagging *all* of the words in a document—even discounting stopwords—tends to result in displays that are overwhelming and often uninformative (see Figure 3.6). Worse, tagging all words equally can sometimes be *negatively* informative.


Figure 3.6: Top: Traditional color-coded tagging creates an overwhelming view that is difficult to read and more difficult to interpret correctly. Bottom: Using our system of ramped tags, the most important words stand out, and the entire passage is easier to read.

For instance, there may be some words that have too low a frequency for LDA to "know what to do with," yet must get *some* tag: likely just the most common one around them. Researchers accustomed to the practice of close reading may read too much into these relatively less "meaningful" tags. As such, we need to sparsify the display to give greater perceptual weight to words that the model deems more "important." And for those readers who are particularly interested in individual words, we need to give them an idea of what *else* any given word might relate to.

There are many ways to define which words are important. Within the toggled-on topics, we deem the "importance" of a given word to be its ranking within its topic, using whatever ranking scheme is currently enabled (saliency being the default—see Section 3.4). We divide words into bins along a single-hue ColorBrewer ramp based on their ranking, giving darker tags to higher ranked words and vice versa. On a white background, this has the double benefit of drawing user attention to meaningful words as well as greatly sparsifying the visual display, making the text easier to read (see Figure 3.6).

Apart from letting readers focus on the most salient words, this method of tagging also conveys the inherent uncertainty associated with probabilistic methods

like topic modeling. This is often difficult for readers to accept. Such was our observation from our work with humanities scholars. Our collaborators would often focus in on one surprising word, perhaps exclaiming: "Why is *that* in Topic 3?!" Sometimes the answer might be meaningful: that word is actually associated with others in Topic 3 in an interesting and surprising way. However, the answer might not be meaningful. Maybe the word appears in *every* topic at some point, or perhaps it is seen so infrequently that the model did not have enough context to informatively tag it, or it may just be the luck of the draw. By using saliency-based color ramps, readers focus much more on the *meaningful* words, as determined by the model. Additionally, conveying this element of uncertainty within the model lets new users begin to appreciate the inexact nature of the algorithm. This decreases their tendency to accept each aspect of a model as gospel, or to dismiss entire models out of hand when they exhibit some unexpected property.

Given our readers' predisposition to focus on single words or groups of words, we needed to go one level deeper beyond the level of passage. Clicking on any individual word, then, takes the reader into the deepest level of Serendip, RankViewer, allowing them to see how that word—and any others in which they might be interested—are dispersed throughout the topics (see Section 3.6).

## 3.6 Viewing words

Individual words are not typically the focus of exploration within a topic model, but they are frequently the objects of study within the humanities. They can also serve as an accessible entry point to a model within *any* domain. Even barring thorough knowledge of topic modeling, most any researcher will be able to come up with a few words whose behavior they would be interested to track within a corpus.

Single words also offer an additional form of adjacency within the topics, and thus another opportunity for serendipitous discovery. While watching our collaborators use our tools, we saw that their interest was often piqued by "surprising" words—words appearing in a topic the user thinks he or she understands, but which do not immediately fit with that understanding. As described in Section 3.5, there are many reasons why such surprises may occur—some interesting and some not—so it is important to filter out the less meaningful ones using saliency ranking. But for the salient surprises, the user's immediate question tends to be "What *else* is that word in?"

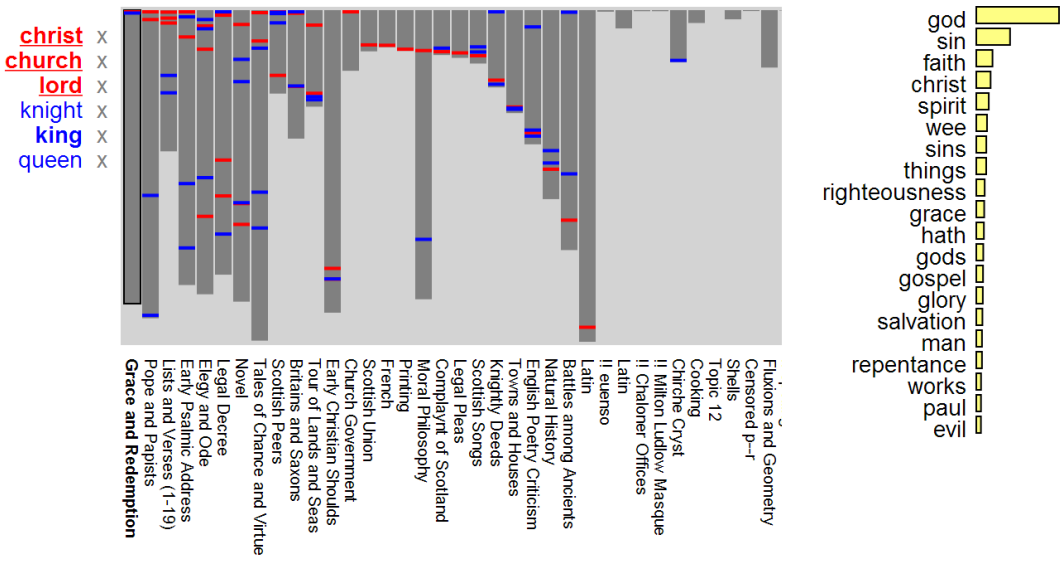



Figure 3.7: RankViewer shows where words fall in the rankings of individual topics. Topics are represented by gray bars and can be sorted by any combination of words being searched for (which are underlined). Individual lines indicate each word's ranking within the topics, color coded to match the list on the left. The view on the right displays the top-ranked words of a selected topic.

RankViewer (Figure 3.7) was created to answer this question. A simple list of topics containing a word is insufficient, because the saliency of a word within a topic (*where* it falls within the topic) is important for determining its relevancy. Instead, our tool shows where a word—or group of words—appears within topic word rankings using lines in an inverted bar chart. Gray bars indicate the relative size of the topics, which can vary dramatically within a model. Color coded lines within these bars correspond to the ranking of individual words that the user has indicated for analysis, either by clicking on them in TextViewer or CorpusViewer, or by manually searching for them within RankViewer. Topics can be sorted and rearranged based on the prevalence of a particular word or set of words, and clicking on a given topic creates a fuller display of its rankings on the right.

This level of depth lets users confirm the importance of words within a topic, improving the validity of their interpretations and strengthening their understanding of the model. By juxtaposing topics in a different way, RankViewer also opens up new pathways for exploration at higher levels. After seeing that an interesting word is present in another topic, users can move quickly to that new topic in CorpusViewer to explore it in depth, examining other documents which contain it. This cascading effect is productive, allowing for bottom-up exploration, and ensuring that inquiries don't necessarily "bottom-out" at the lowest (word) level.

## 3.7 Implementation

Our prototype for Serendip is web-based so as to make it easily shareable and accessible to a variety of users. Serendip operates on a back-end written in Python with the Flask framework and a front-end written in Javascript and D3 (Bostock et al., 2011), with Twitter's Bootstrap providing UI elements. Topic model data is stored on the server as CSV files, and texts are stored as lists of tokens, also in CSVs. The models described in our use cases were generated using Mallet (McCallum, 2002).

There were a number of engineering challenges that were necessary to overcome in order to allow users to interact with full-scale models and large documents within the browser with little latency. Within CorpusViewer, larger models (more than a thousand documents) would initially operate very slowly as a result of the huge number of elements being created within the Document Object Model (DOM): a line for each topic and document, and a circle at each vertex. While we were able to scale up slightly by filtering out vertices that fell below a particular proportional threshold, models containing tens or hundreds of thousands of documents were still prohibitively slow. However, the vast majority of DOM elements slowing down interaction were off-screen, and not visible to the user. We ultimately implemented a dynamic process for building only the parts of the DOM that the user could see. Off-screen elements beyond a certain buffer are only created once the user scrolls close enough to them, or once a re-ordering demands that they be brought on-screen. This buffering technique does not change the user's experience, as elements are built right before the user scrolls enough to see them appear, but allows users to interact with much larger models smoothly.

We had a similar issue when attempting to present entire novel-length documents to the user for perusal within TextViewer. We initially implemented the passages of tagged text within TextViewer as pre-computed HTML spans, but these slowed down interaction in much the same way that the DOM elements did in CorpusViewer. This approach was replaced again by an approach of just-in-time DOM building. Rather than pre-computing all of the HTML, documents are loaded in the browser as lists of tokens, which can then be sliced to retrieve the tokens meant to be visible to the reader at any given time. As the user scrolls around the document or navigates using the line graph overview, we are able to create the proper HTML spans as requested, eliminating the latency associated with exploring large documents.

## 3.8 Use cases

We describe here some experiences using Serendip on various corpora. The first use case was performed by visualization researchers, and is intended to illustrate the capabilities of our techniques on a familiar dataset. The second use case was performed by a literature scholar with experience using the tool and provides an example of serendipitous findings on real data. Other initial use cases with domain researchers (not reported here) include a collection of over 600 plays, and a large collection of novels.

### 3.8.1 Vis Abstracts

To demonstrate Serendip's features, we used it on a familiar corpus: a collection of abstracts from select IEEE sponsored visualization conferences from 2007-2013, including SciVis, InfoVis, VAST, BioVis, and PacificVis. We standardized the conference names and used various heuristics to remove bibliographic entries for "non-papers." The corpus consists of 1127 abstracts, ranging from 30 to 389 words. The discussion below is based on a 30-topic model. The findings on a familiar data set help us confirm that system features provide reasonable results.

We started with a common question: are the content differences between the conferences reflected in the model? For an initial look, we chose a 2D spatial embedding (using Spectral Embedding), coloring the scatter plot by conference (Figure 3.8). This provided a picture with the thematic conferences being relatively distinct, while the general conference (whose topics span the range of the others) is more spread out. To see which topics create the distinction, the ANOVA ranking was used to order the topics. The top topic has terms *visual, analytics,* and *analysis* among its top terms, and showed that many papers from the Visual Analytics (VAST) conference self identify themselves in their abstracts. The second ranked topic also identified VAST papers, using terms related to a design challenge hosted by the conference. The third topic featured the terms *volume* and *rendering.* The lowest ranked topics for distinguishing venue featured generic terms, such as *problem* and *approach.*

Next, we re-ranked the topics based on their ability to contrast VAST and InfoVis papers. Amongst the least distinctive topics were not only topics with generic terms (e.g. *problem, approach, ...*), but also a topic featuring *time* and *dynamic* and one with *space* and *dimensions,* both suggesting common topics at the venues.

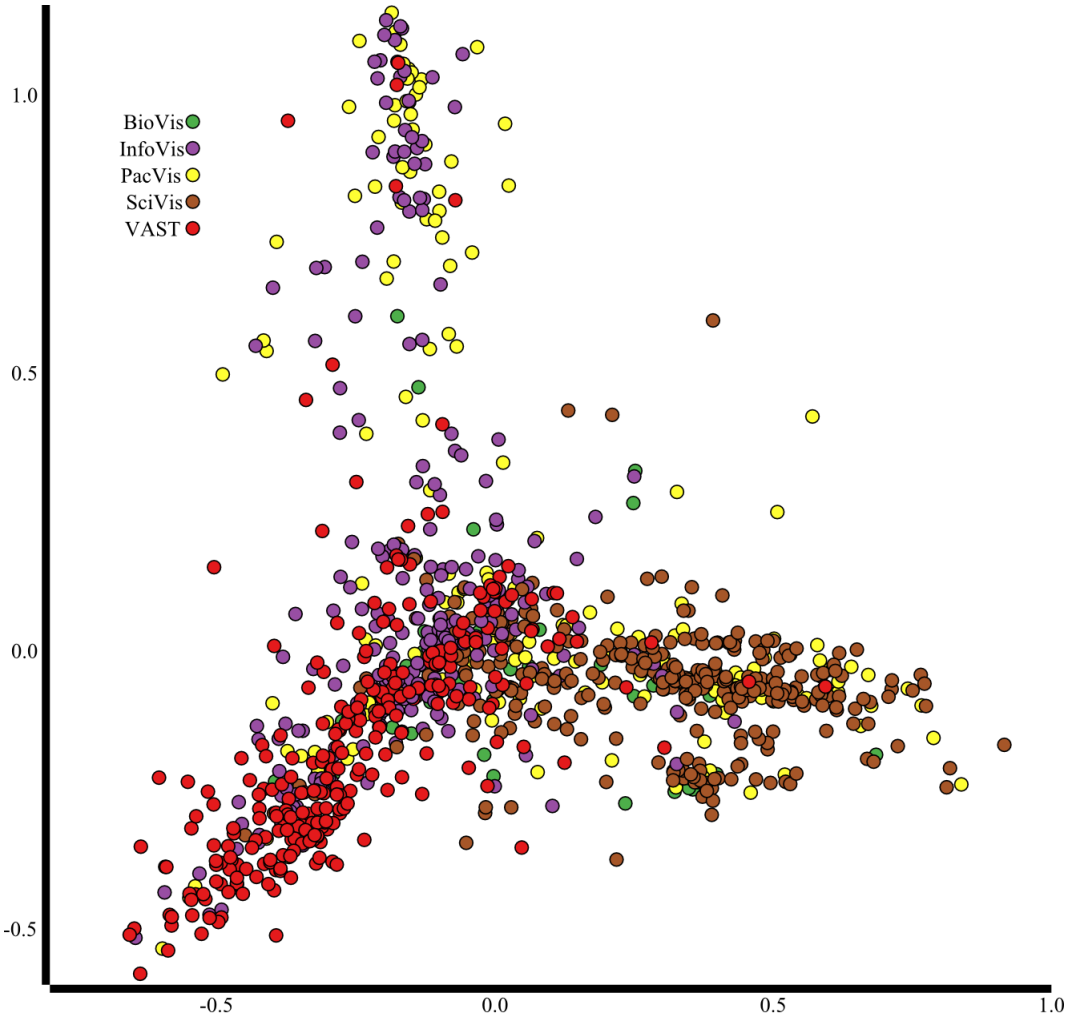



Figure 3.8: A scatterplot of an embedding of the documents in the VisAbstracts corpus. Spectral embedding was applied to the document vectors. Each point represents a document, and is colored based on the venue of the document. The plot shows, at a glance, that the topic data is capturing some sense of the distinctions in the venues. Venues with more focused themes (VAST, InfoVis, SciVis), tend to group more closely together, while general venues (PacificVis) are more diverse.

Ranking topics by their ability to contrast 2007 and 2013 indicated topics that have changed. The two highest ranked topics for contrasting these years were *studies, significant, evaluate, ...* and *fast, gpu, ...* While many of the lowest ranked topics were generic terms, some recurring challenges, such as *imaging, diffusion* also appeared.

For a more specific exploration, we sought to look for documents that might be similar to our own work. We began by picking a single paper that came to mind as similar to the material in this chapter: ParallelTopics (Dou et al., 2011).

This paper had only a single salient topic, but one that was clearly relevant ("text, search, learning"). To create a broader query, we searched for all papers with the string "topic" in their title, and ranked by distance to the resulting set of 5 papers. When using the "distance to group center" ranking, the 5 papers were *not* closest to the center—in fact one of the papers was not in the top 30. The top ranked documents shared some aspect of the topic visualization problem, such as handling uncertainty, but not all discussed text visualization. Among the top documents were many relevant papers, including at least two of which we were not previously aware but now are discussed in our related work. Using the minimum-distance-to-set ranking did place the group at the top of the ordering, and put a slightly different, but still relevant, set of papers next to them.

### 3.8.2 Early Modern Literature

Our collaborators have developed a corpus of 1080 digitized texts published between 1530 to 1799. The corpus was built by randomly sampling 40 texts per decade from a larger archive, in an attempt to provide a less biased view than just using well-known texts. However, this means that the corpus has significant diversity and is unfamiliar to most who work with it. With documents ranging from a few hundred words to hundreds of pages, the corpus is too large for any researcher to read manually, and so we've been interested in seeing how the task of exploring it scales within Serendip. This documents have been run through the VARD 2 modernizer (Baron et al., 2011) and annotated with metadata such as year, genre, author, publisher, etc.

A literature scholar (our collaborator, Mike Witmore) spent an extended period of time exploring the model over a number of weeks, working between the tool and the texts themselves. He began with the topics themselves, assigning names based on his sense of the texts containing the topics and the distribution of topic words. Naming required extensive switching between levels and combining various sources of information. It also led to some surprises. For example, using the topic lists and RankViewer, he observed that there were a handful of topics containing long lists of numbers. Some contained numbers from 1 all the way up to 100. Another only contained the numbers up to 20 or 30. Examining the documents containing this latter topic revealed that many were written by Protestants. Drilling into the passages revealed the reason for the cutoff: the authors were giving references to Bible passages to support their arguments, and

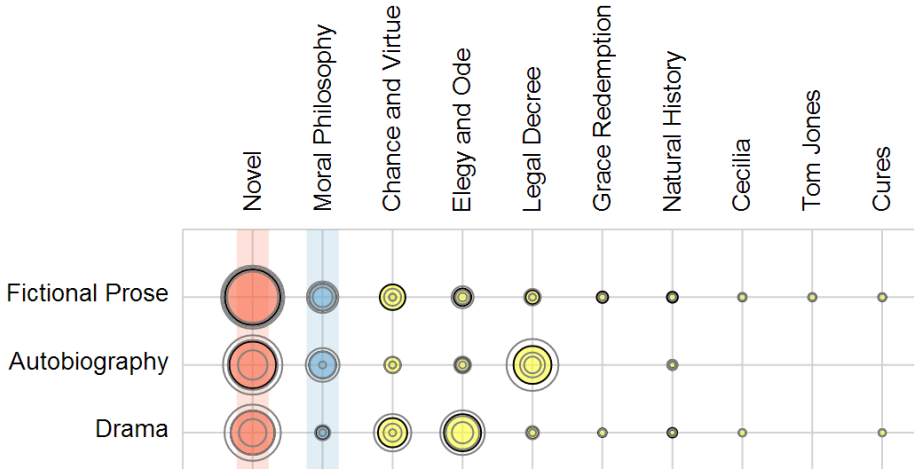



Figure 3.9: Sorting topics by the aggregate genre "Fictional Prose" creates an unexpected juxtaposition of topics concerning the novel and moral philosophy.

the numbers 20-30 were distinctively biblical (as opposed to 1-20, which were spread across the corpus more broadly). The scholar named the topic "Grace and Redemption."

After building familiarity with the model, the scholar continued to explore it, combining multiple information types and levels of scale in ways that not only answered posed questions, but led to serendipitous discoveries as well. For example, he was able find support for the argument—advanced by scholars of the novel—that the English novel was the literary expression of English moral philosophy (ethical works designed to guide the conduct of citizens). This exploration began by aggregating documents by genre and honing in on a particular one labeled "Fictional Prose" (by a human bibliographer). After sorting the topics by this genre, the top two were topics with which he was familiar from earlier explorations and had labeled "Novel" and "Moral Philosophy" respectively—an interesting juxtaposition (see Figure 3.9). Sorting the topics by similarity to one another confirmed that these two were very closely related. By drilling into the prose fiction genre within CorpusViewer, the user identified a few texts in which to look for examples of this overlapping or convergence. He began with Samuel Richardson's *Pamela*, one of the first English novels.

From a passage of the novel, he was able to assess how words from both topics were interacting: words from the "novel" topic were introducing concrete characters whose actions were the subject of moral evaluation, while the words from the "moral philosophy" topic were applying abstract concepts to those actions, concepts that are the main subject of more argumentatively rich ethical writing

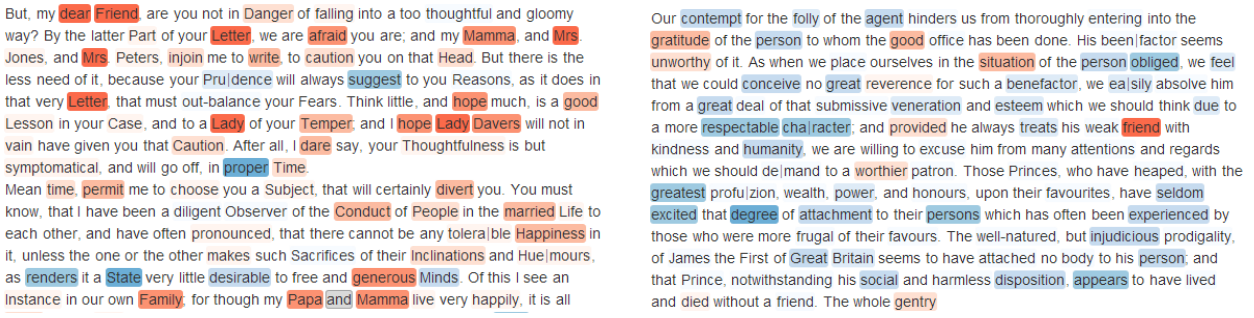



Figure 3.10: Passages from the novel *Pamela* (left) and the *Theory of Moral Sentiments* (right). The topic associated with novels is shown in read, while the "moral philosophy" topic is shown in blue.

(see Figure 3.10). That convergence of these two kinds of topic words made sense to the user, since the novel must not only analyze actions (involving the reader in a parallel exercise of active moral evaluation), but also render those actions in a rich narrative. In a work of moral philosophy, Adam Smith's *Theory of Moral Sentiments*, the scholar was then able to explore the pattern in reverse (see Figure 3.10). In this document, words associated with the novel were interwoven with argumentation about moral sentiment and conduct, a finding that also made sense, since moral philosophy must—perhaps unlike metaphysics or logic—take its cues from concrete human action. In other words, there can be no novel, nor any moral thinking, without a concrete and specific situation of personal action and deliberation. Focusing in on the word "situation," the user then transitioned into RankViewer and found that this word, which is often used to shift readers away from their immersion in the story into a more explicitly evaluative cognitive frame, was highly rated on both topics.

The analysis had thus progressed unexpectedly through four levels of abstraction: a ground truth had been correlated with algorithmically generated topics (a topic, "novel", tracked reliably with works aggregated as prose fiction); that ground truth was then extended into an unexpected juxtaposition (the close relation of the "novel" and "moral philosophy" topics); exemplary works were identified and their narrative techniques evaluated on the level of passages and individual words (novel and moral philosophy words intermix on the page); and finally, topic words were found that sit at the intersection of these two narrative forms ("situation"). Having opened up a new level on which to explore a current critical debate about the novel, our user then returned to the matrix view to rate the existing genres according to their scores on the "novel" topic—an exploration that was also

suggestive, since "novel" captured not only prose fiction, but texts classified as autobiography, drama, travelogue, and biography. Each of these sub-genres has also been related to the novel in literary studies, and so the user was able to begin generating hypotheses about how novelistic language might have developed from, or be shared with, these types of writing, many of which pre-date the novel as literary forms.

## 3.9 Discussion

Serendip was intended to support and promote scalable, multi-level serendipitous discovery in text corpora, a task at which we believe it has succeeded after observing its use by a variety of collaborators. We believe that the tool was able to achieve its goals because it multiplies the angles from which users can enter and then transition through a corpus—in effect, minimizing "roads not taken." Our use cases show investigations that combine topic models with other sources of information to reveal discoveries at multiple levels of detail. Our methods for addressing scale seem to apply for the corpora with thousands of documents, texts on the order of full books, and hundreds of topics in the model. The various starting points and ways to use intermediate results and questions to springboard to next steps provide a fluency of exploration that keeps users engaged.

One drawback of Serendip's approach is the static nature of the models upon which it operates. While Serendip assumes that the model has already been created when the researcher begins their analysis, there are many places within the model creation and training process where the researcher's input and knowledge could help result in a better model. Chapter 4 attempts to address this issue by providing techniques for the comparison of *multiple* topic models that should help researchers select the correct model to use based on the questions they wish to ask of it.

A more technical challenge is developing a more rigorous mathematical toolkit for working with document vectors. As mentioned, our current distance metrics, averaging methods, and statistics do not preserve the sparse, convex structure of the vectors. Local neighborhood graph distances seem promising. While Serendip offers a rich set of ordering metrics, an improved set would offer better opportunities for achieving scaling through placing relevant objects first, and serendipity by bringing different things together. One promising avenue is to apply distance metric learning approaches to allow users to craft ordering functions

based on sparse sets of examples.

In the future, it would be beneficial to add improved methods for finding passages, e.g., through similarity to exemplars. Support for a richer array of model types, such as hierarchical or multinomial topic models would also be an interesting extension (Dou et al., 2013).

By providing a set of visual encodings that address multiple levels of exploration going from distant to close reading, interaction techniques for coupling these views, and statistical techniques for linking different sources of information, Serendip allows users to use topic models and other information to guide the exploration of text corpora. The methods are designed not only to address scale, but also to promote serendipitous discovery. Our experience using the tool for literary scholarship suggests that it engages users and helps them make discoveries.

# 4 COMPARING TOPIC MODELS

*O Helen, goddess, nymph, perfect, divine!*
*To what, my love, shall I compare thine eyne?*

— Demetrius, *A Midsummer Night's Dream*

In Chapter 3, I described a set of techniques that can be used for exploring a single topic model, along with a system called Serendip that I created to afford such exploration. However, as discussed, one of Serendip's key limitations is its lack of support for 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. 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. In existing literature, these comparisons are generally 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 (Chuang et al., 2013; Stevens et al., 2012).

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, it is important 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, many of which came up in Chapter 3. 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 have composed 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.

The main contributions contained within this chapter 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 novel designs including **buddy plots**, a novel way of looking at relative distances within a corpus.

Work contained in this chapter was originally published as Alexander and Gleicher (2015).

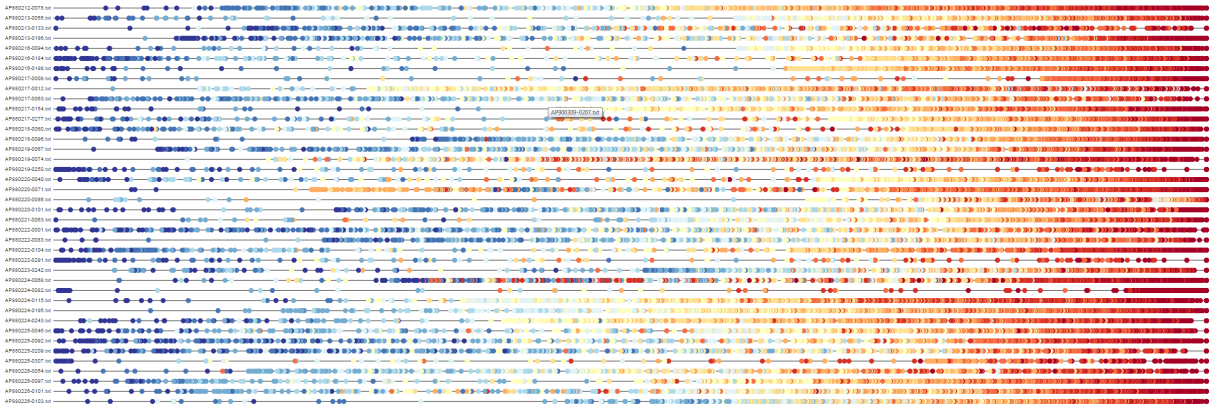


Figure 4.1: Buddy plots show consistency of document relationships across topic models by encoding similarity with respect to individual documents. In this figure, each row represents a document, with the rest of the corpus encoded as circular glyphs along the row. Distance from the row's document in 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.

## 4.1 Related work

The process of topic modeling and existing techniques for visualizing single models of text corpora are discussed in Chapter 2. This section will touch on work related

specifically to topic model comparison.

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 (Wallach et al., 2009b), often disagreeing with human judgments of the documents (Chang et al., 2009). Chang et al. 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 (Chang et al., 2009). 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 (Newman et al., 2010) and document co-occurence within the original corpus (Mimno et al., 2011). However, these may disproportionately favor certain modeling types (Stevens et al., 2012). AlSumait et al. 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 (AlSumait et al., 2009). 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 (Chuang et al., 2013; Stevens et al., 2012).

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 (Chang et al., 2009). Chuang et al. compare topics to expert-generated categories to filter them into classifications of *junk*, *fused*, *missing*, *repeated*, or *resolved* (Chuang et al., 2013). 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. use matrix views to compare topics to ground-truth concepts generated by a group of experts (Chuang et al., 2013). 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. use bipartite graphs and two-dimensional embeddings to compare models built with LDA and LSA (Crossno et al., 2011). 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 4.3.1).

## 4.2 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, I will outline these decisions and tasks, and describe why we believe they require a different set of model comparisons than is currently available.

### 4.2.1 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: as discussed in Chapter 2, a variety of topic modeling techniques exist. These model types have been show to perform differently depending on the task (Choo et al., 2013; Crossno et al., 2011; Stevens et al., 2012).

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. 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 (McCallum, 2002), they can have a significant impact on topic legibility (Wallach et al., 2009a).

Though there has been some work visually exploring this hyperparameter space (Chuang et al., 2013), 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 importance to the analyst but may be common enough to dominate statistical analysis. This can include articles, pronouns, and other function words, but can also 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 (Hope and Witmore, 2010).

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 be challenging to define. 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 4.4). 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.

### 4.2.2 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. Topics are often represented by their top three words (Chaney and Blei, 2012), word clouds (Xu et al., 2013), or ordered lists (Chuang et al., 2012a)—evaluations of competing encodings are described in Chapter 5.

**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 (Alexander et al., 2014; Chuang et al., 2012b).

**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 (Hu et al., 2012), 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 (Cui et al., 2011; Havre et al., 2000; Wei et al., 2010).

Each of these single-model tasks corresponds to a type of comparison that a user might make between models (Table 4.1). The primary comparison task involved in understanding topics is **topic alignment**: how well do the topics from the two models match? If they don't, why not? 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.

Table 4.1: Mapping Single Model Tasks to Comparison Tasks

| **Single model task** | **Model comparison task** |
|---|---|
| Understanding topics | Topic alignment |
| Understanding similarity | Distance comparison |
| Understanding change | Timeline comparison |

## 4.3 Visualizing comparison

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 potentially topics on the order of hundreds, documents on the order of 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.

### 4.3.1 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, including:

- *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.

**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).

We can re-order topics to best expose the matching structure. While a number of potential algorithms are possible (see Mueller et al. (2007) for a survey), we

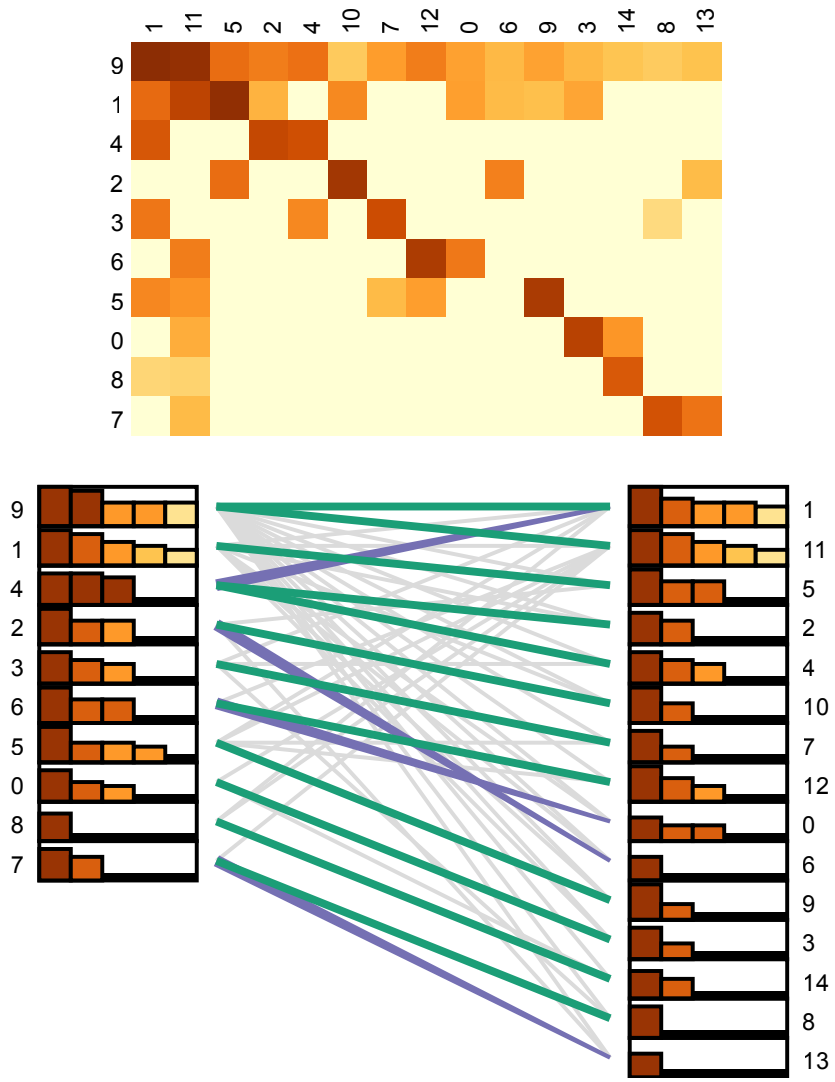



Figure 4.2: Topic alignment between two models built on the works of William Shakespeare, one with 10 topics and one with 15 topics. On top, a heatmap of topic alignment indicates which topics from the two models are closely matched (dark orange indicating a close match, yellow indicating no match). Below, a bipartite visualization indicates matches of different strengths (green as a two-directional match, purple as a one-directional match, and gray as a weak match) and the bar charts next to each topic show the strength of the top five matches (with bar height encoding strength and color used to show rank so that ties are salient). Topics exhibiting multiple close matches (e.g. Topic 4 in the 10-topic model) may be instances of merged concepts to explore more closely.

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 4.2). 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 filtered using a threshold, or omitted completely. As it is difficult to encode link strength on the lines, we augment the parallel coordinates with a small pareto bar-chart next to each topic's node. The bar chart shows the strength of the top (usually 5) matches, and color is used to encode rank (with similar values counted as the same) so that cases where there are ties are clearly visible. The stack of colored pareto bar charts allows for quick scanning to identify whether each topic has a strong best match, and whether this match is clearly the best.

We also allow users to look at subsets of the topics—e.g., just the topics that are shared between two documents of interest. This can be helpful for understanding how and why individual document relationships change between models. An example of this sort of comparison is described in Section 4.4.3.

#### Topic comparison

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 4.4.4), so we should not 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 4.3). The visualization takes an *object* topic from one

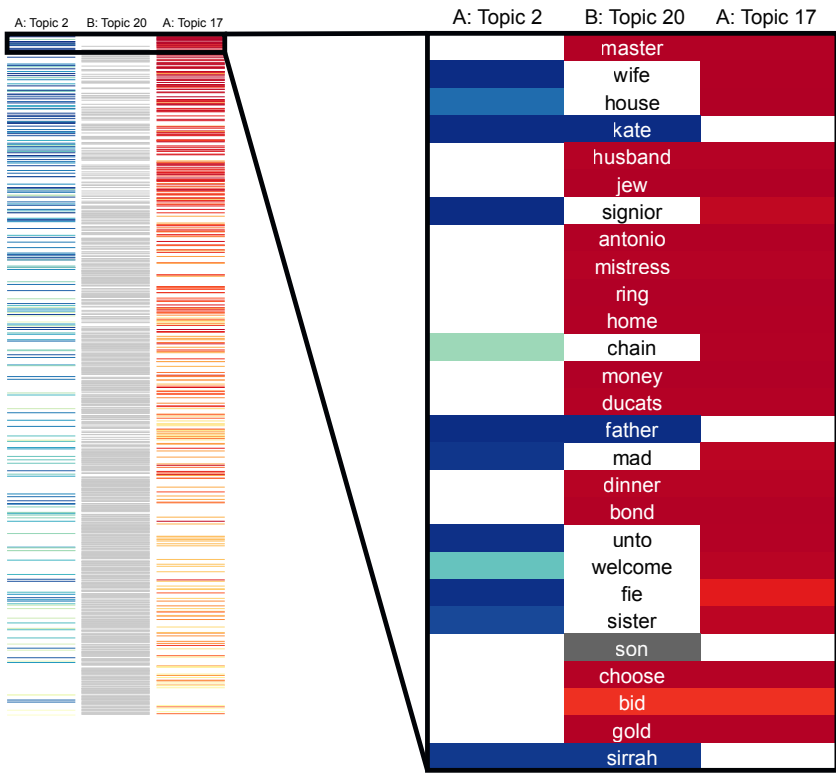



Figure 4.3: Two topics from one model (Model A) built on the works of Shakespeare split into 1000-word chunks (topics $2_A$ and $17_A$, colored with blue-to-yellow and red-to-yellow ramps in descending order by frequency) to a topic from a non-chunked model built on the same texts (topic 20 from Model B, unique words colored in gray). The overlap of words within topic $20_B$ seem to possibly indicate a merged topic. Zooming in to see the actual words in the window to the right shows a semantic difference between $2_A$ and $17_A$: $2_A$ seems to be related to family while $17_A$ seems to be more about wealth.

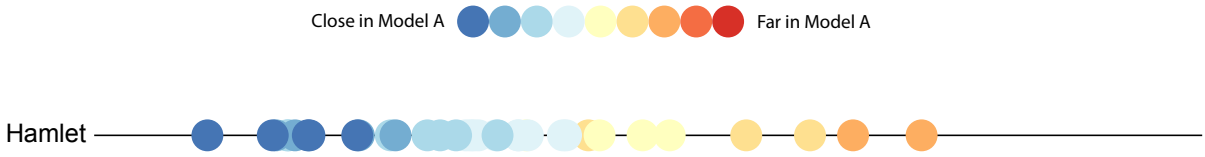
model and potentially multiple *reference* topics from the other model. For each reference topic, we create a color transfer function encoding words with colors from a continuous ramp—interpolated from a discrete ColorBrewer ramp (Brewer et al., 2003)—based on their rank within the topic. We then use 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 4.3. To evaluate these splits, we have a moveable window allowing users to zoom into the individual words. For example, in Figure 4.3, 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.

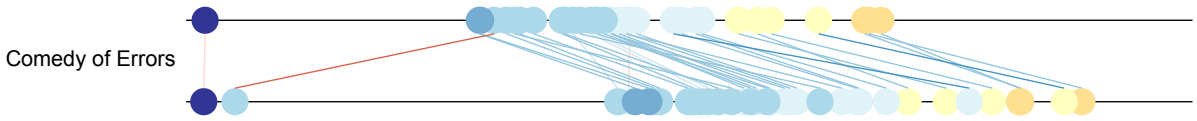
### 4.3.2 Understanding similarity

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 (AlSumait et al., 2009) and rescaled dot product (Chuang et al., 2013).

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.




(a) In this example, distance within a 25-topic model built on the works of Shakespeare is encoded as horizontal position (closer documents to the left, further to the right), while distance within a 50-topic model built on the same documents is encoded as color along a ramp going from blue (close) to red (far).




(b) In this parallel buddy plot, distance in the 50-topic model is encoded along the top axis while distance in the 25-topic model is encoded along the bottom axis. (Color is consistently encoded on both top and bottom as distance in the 50-topic model to facilitate comparison.) Most documents are further from *Comedy of Errors* in the second model, with one easily identifiable exception (*Taming of the Shrew*).

Figure 4.4: Buddy plots encode the distances of corpus documents away from an individual *reference* document (labeled to the left). By using both position and color, buddy plots can combine multiple sets of distances within a single line (Figure 4.4a). They can also be used in parallel (Figure 4.4b), using two axes to show precise movement between two models.

**Buddy plots**

A common way of looking at document distance is to place documents into a two-dimensional embedding (Crossno et al., 2011). 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 perspective 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, typically encoding one set of distances with horizontal position and the other with a diverging color ramp (Figure 4.4a).

**Parallel buddy plots** have two axes, encoding each set of distances on its own axis and connecting them with parallel-coordinate-style edges (Figure 4.4b).

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

(Albers et al., 2011), drawing the user's attention to outliers that can be inspected further or explicitly zoomed in on (see Figure 4.5). 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.

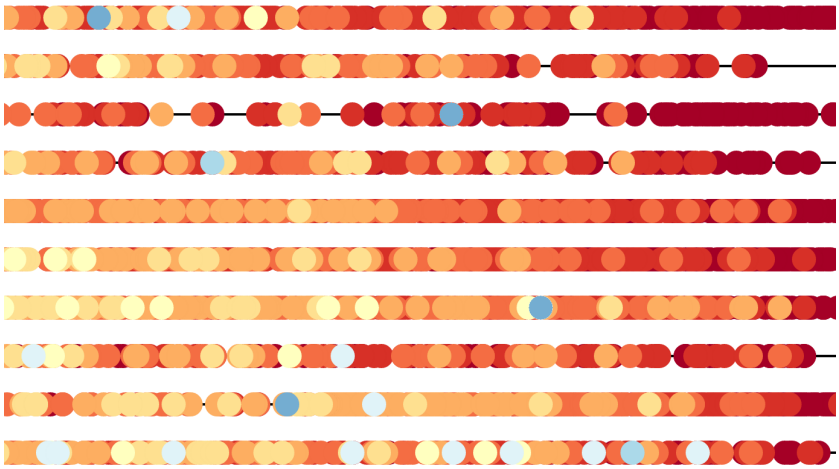


Figure 4.5: Zooming in on buddy plots can help pick out individual documents. Also, by changing draw order, we can make sure interesting outliers are easily found. In this example, draw order is controlled by the amount which a document has moved from one model to the other. Though there are a large number of red documents overdrawing one another, we are easily able to pick out the blue documents, which were close to the row's reference but moved much further away. Clicking on one brings up its metadata, allowing users to compare its specific topics to those of the reference, as in Section 4.4.3.

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 4.1). Patterns of parallel buddy plots can show directional or density trends from one model to the next (Figure 4.13). Though single buddy plots are more space efficient, it is still difficult to fit an entire corpus onto the screen. As described for single models in Chapter 3, 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 of 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 $r_{rank}(n, p)$ 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 4.6. 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.

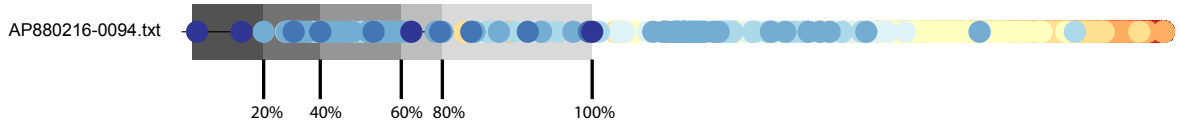



Figure 4.6: *Pareto radii* are used to indicate how much a document's "neighborhood" spreads out from one model to another. Here, the gray gradient shows pareto radii in a trimmed model built on AP news documents that are needed to cover 20, 40, 60, 80, and 100 percent of the closest 20 documents to 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 4.7). 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 4.5). Other potential draw orders include rank within either model, as well as change in rank or distance.

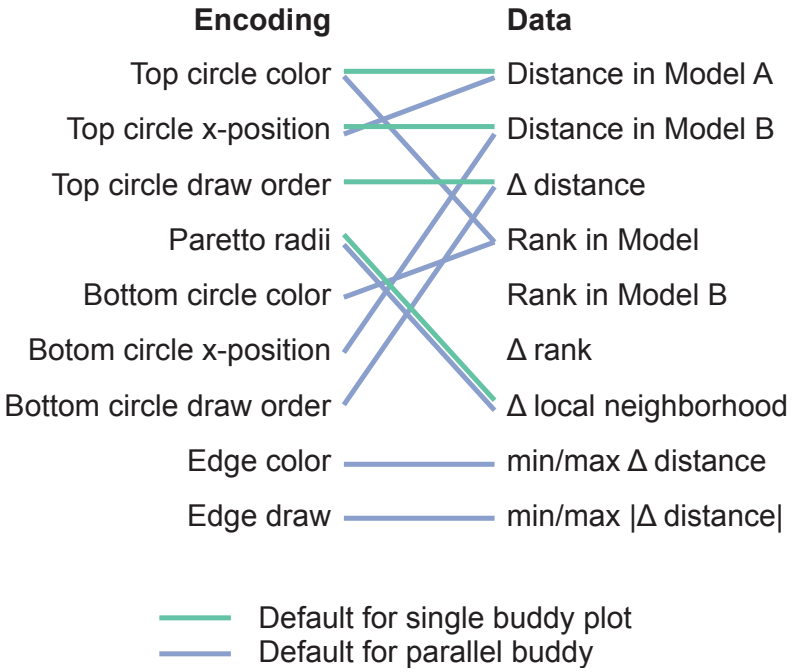
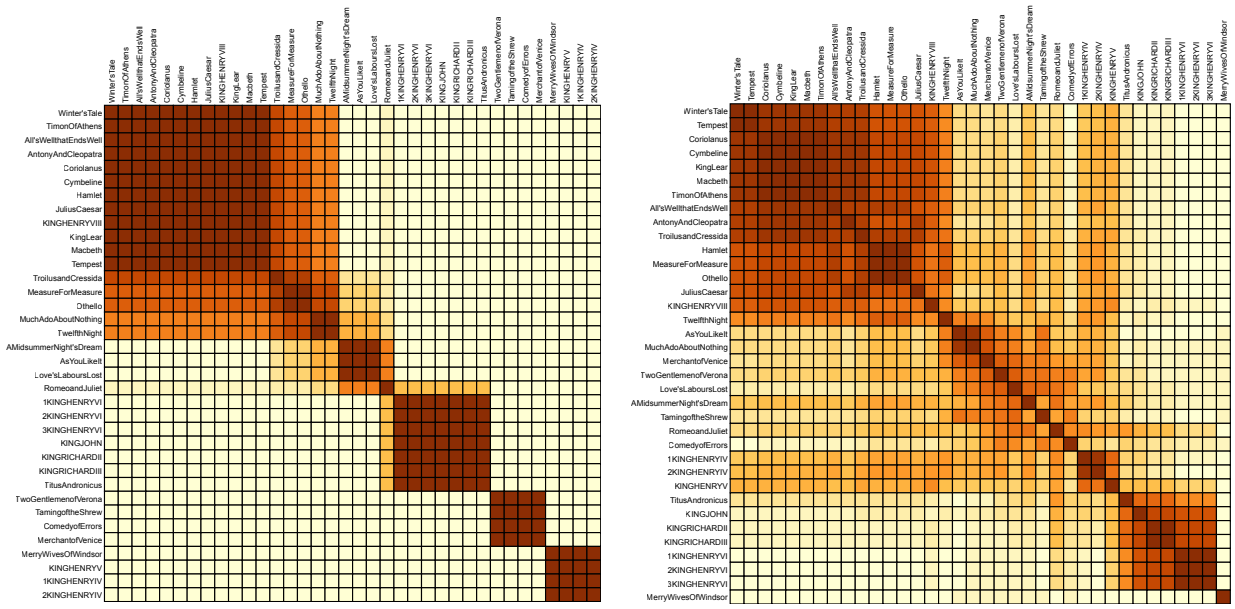



Figure 4.7: Buddy plots allow researchers to explore comparisons by combining a variety of encodings and data. Though most data and encodings can be combined, we have a set of defaults that seem well suited to single and parallel buddy plots.

### 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 (Filippova et al., 2012). 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.




(a) Model with 10 topics

(b) Model with 50 topics

Figure 4.8: These heatmaps show the consistency of clusterings resulting from 20 runs of k-means on two models built from the works of William Shakespeare (k = 5). Darker brown squares indicate pairs of documents that consistently appear within the same cluster. Lighter yellow squares indicate pairs of documents that never appear within the same cluster. Here, the 10-topic model seems to have more consistent clustering behavior.

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 4.8. 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.

### 4.3.3 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 (Cui et al., 2011; Havre et al., 2000; Wei et al., 2010). Though there are many variations, such diagrams typically have individual colored bands for each topic. The widths of these bands encode the portion of the corpus represented by a particular topic at a point in time (with time represented as position from left to right 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 4.9). 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 4.3.1). 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 4.10). This distinction can help the user decide whether the split is a semantic one or a random artifact of the model parameters.

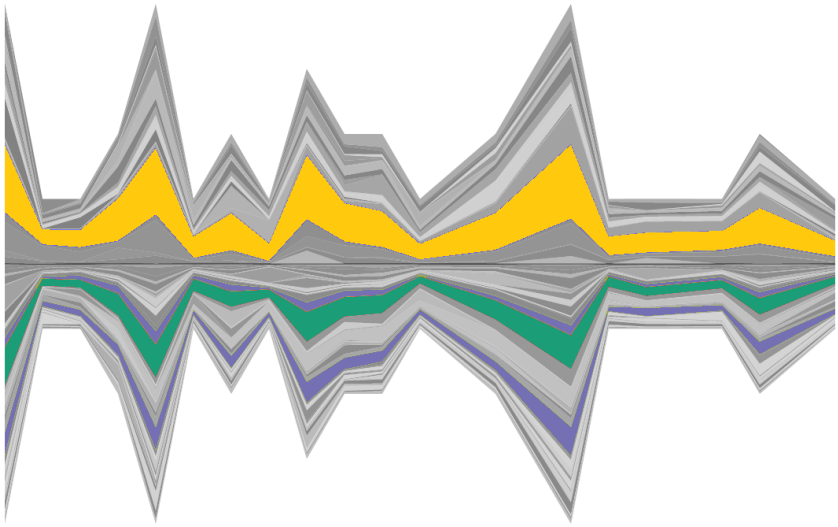


Figure 4.9: Asymmetrical topic flow diagrams show how topics change over time within two models. The horizontal axis encodes time; width of bands above the axis indicate topic proportions from one model while those below the axis indicate proportions from the other model. Hovering over an individual topic highlights it in yellow and highlights any aligned topics from the other model as in Section 4.3.1 (green for a two-way match, purple for a one-way match).

(a) Some splits happen uniformly across time

(b) Some splits alternate with events

Figure 4.10: Asymmetrical topic flows allow the user to select individual topics, filtering out others so as to compare them to their best aligned matches.

## 4.4 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 (Wang et al., 2012). All models were built with MALLET (McCallum, 2002).

### 4.4.1 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 (Choo et al., 2013; Lancichinetti et al., 2015). 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 similarities. Though the differences in topics may fall within a reasonable allowance (see Figure 4.12a), changes in document similarity tended to be much more dramatic (see Figure 4.11). Though using MALLET's built in hyperparameter optimization improved the topic alignment (see Figure 4.12b), 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.

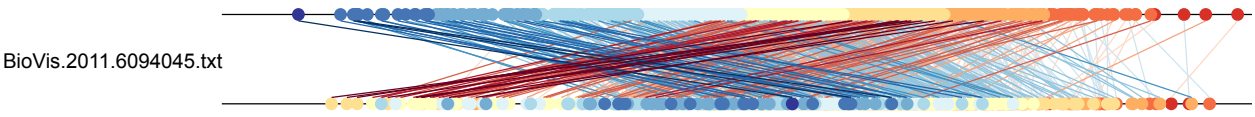
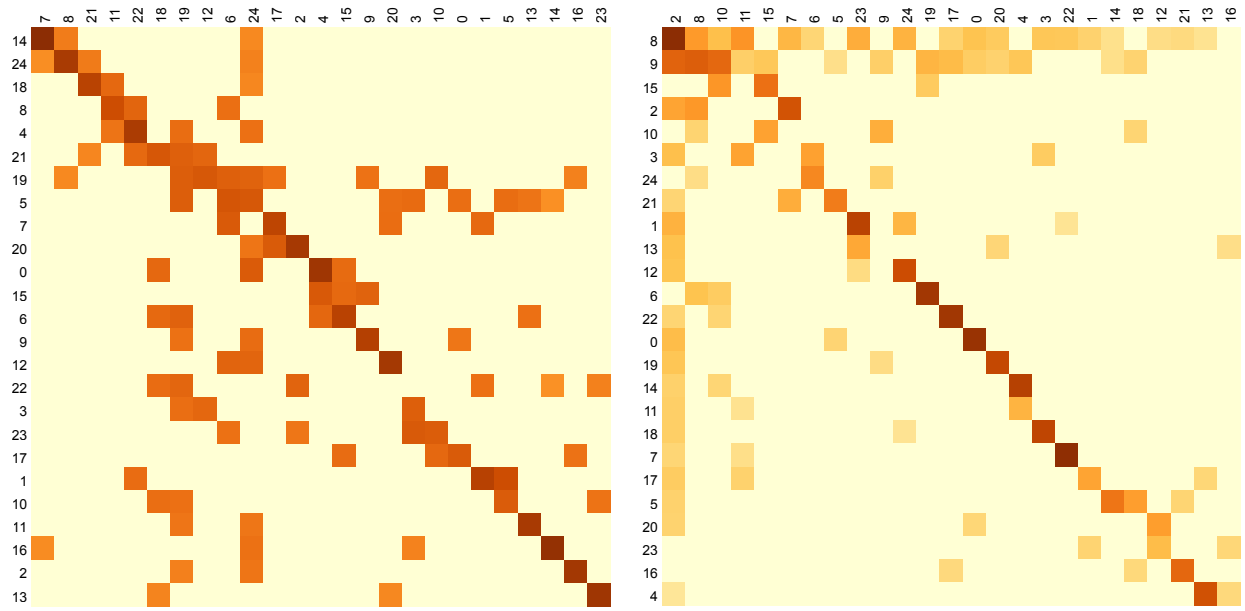



Figure 4.11: This parallel buddy plot compares distances between two 25-topic models built using the same (default) parameters on a corpus of 1127 visualization abstracts. Color encodes distance in the first model for glyphs along both axes for consistency, and edge lines are ordered by greatest magnitude of change in distance. We can see that there has been a dramatic shift in similar documents across the two models.

### 4.4.2 Different numbers of topics (Shakespeare)

As mentioned in Section 4.1, 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



(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.

Figure 4.12: 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.

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 4.8a 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 quite consistent. The 50-topic model as shown in Figure 4.8b, on the other hand, exhibits much less consistency. Many documents appear with almost all of the others in a cluster in at least one of the runs. For a user hoping to explore the differences between these groups, the 10-topic model may give them cleaner distinctions to investigate.

### 4.4.3 Document chunking (Shakespeare)

As described in Section 4.1, 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? Intuitively, longer chunks should create more general models, but it can be hard to predict the effects *a priori*.

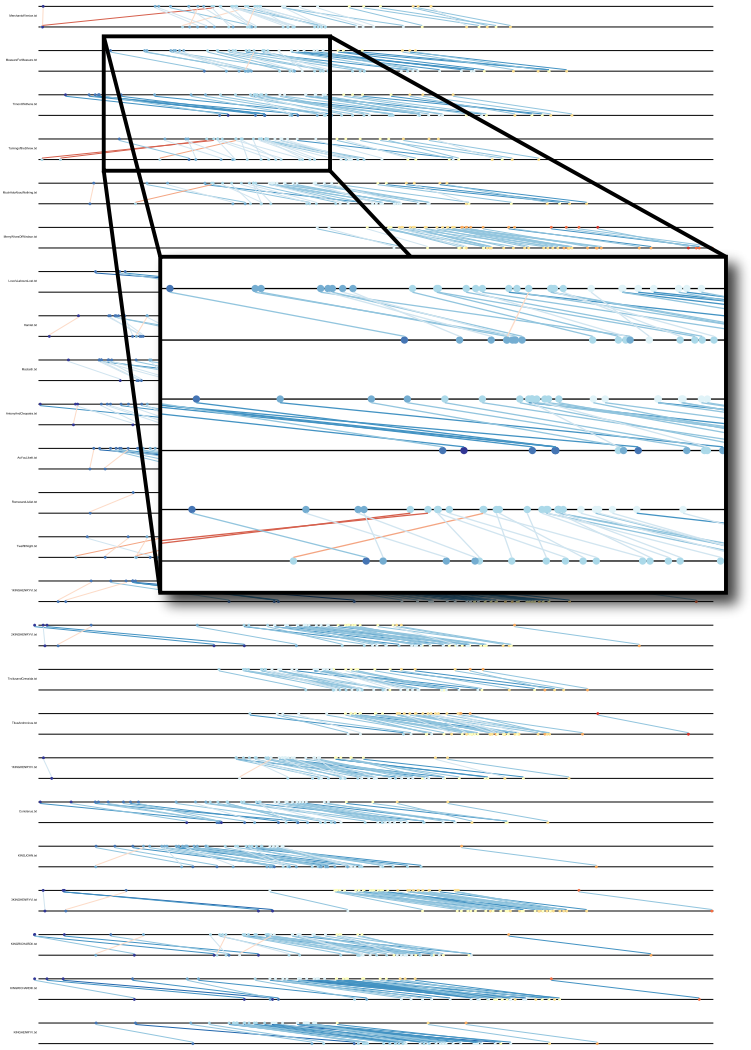


Figure 4.13: Here, expanded parallel plots compare a model built on Shakespeare's works split into 1000-word chunks (the top lines) to a model built on the documents treated as whole (the bottom lines). Red edges indicate documents that move closer in the un-chunked model, while blue edges indicate documents that move further away. From the color and slant of these edges, we see a trend of documents moving *away* from each document, with a few salient red exceptions.

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 Jockers

(2013)—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 (Figure 4.13). 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 4.4b, *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, with the red topic dealing with wealth and the blue topic dealing more with family (Figure 4.3). Depending on the user's interest in these specific topics and documents, incorporating this split may be worth the extra effort of the chunking process.

### 4.4.4 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 (Wang et al., 2012), 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 4.1 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 4.14), we can see that the documents become intensely jumbled.

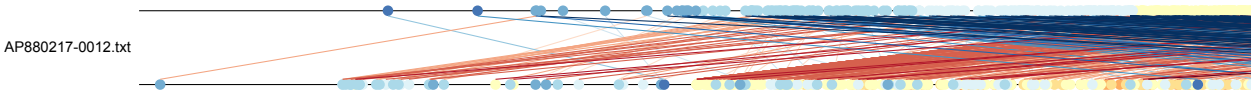



Figure 4.14: This parallel buddy plot compares distances from a 50-topic model built on a corpus of Associated Press documents to distances from that same model with all but the top 50 most frequent words stripped from each topic. (Color encodes distance in the first model. Edges are drawn in increasing order of change in distance.) The act of stripping words from the topics creates a very dramatic change in the document's relationships with the rest of the corpus.

That trimming away such dramatic portions of the topics would have such an effect is perhaps unsurprising. After all, the ability of topic models to semantically relate documents relies on its ability to find subtle patterns of co-occurrence that would likely escape a human observer. However, this emphasizes the importance of the full topic distributions, especially for documents that are low in high-frequency 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.

## 4.5 Discussion

In this chapter, I have argued for the importance of explicit, task-driven topic model comparison. I have offered visual techniques for making comparisons associated with using topic models to understand topics, similarity, and change within text corpora. Among these, I have introduced *buddy plots*, a novel visualization for viewing changes across two sets of distances by shifting our perspective to that of a single reference document. Finally, I have described a number of use cases showing the variety of comparisons afforded by these methods.

In the future, I hope 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, there are also 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 parameter space. There may be a benefit to combining pairwise comparison with higher-level overviews that incorporate more models.

Though we believe that the three categories of understanding topics, similarity, and change cover the vast majority of topic model uses, there are a handful of other tasks to support. For instance, it would be powerful to enable comparison of models at the passage level, using tagged text encodings such as those described in Chapter 3. This could inform users of what different models show about the passages and flow of individual documents.

Topic model comparison is still just a step down the path of offering full interactive model *training* such that domain researchers can input their knowledge into the model building process. Some work has been done on actively tuning models by merging or breaking up topics (Choo et al., 2013), pulling documents apart or closer together (Endert et al., 2012a), or directly intervening in the ongoing iterations of the algorithm (Amershi et al., 2015; Muhlbacher et al., 2014). 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. The sorts of comparisons outlined in this chapter should be useful in identifying which aspects to separate or combine.

Finally, I look forward to applying these techniques to a wider variety of corpora and domains. We have only scratched the surface of potential comparisons to be made, and can use these methods to learn more about things like stopword usage, different modeling types, and models built on non-intersecting sets of documents.

# Part II

# Evaluations of Visual Encodings for Text Data

# 5 TOPIC REPRESENTATIONS FOR GIST-FORMING

*What's in a name?*

— JULIET, *Romeo and Juliet*

In the first part of this dissertation, I focused on systems and techniques for helping researchers use topic models to accomplish high level tasks. Chapter 3 discussed a system and workflow designed to help researchers see how topics formed patterns within large collections of documents, tracing those patterns across many levels of abstraction. Chapter 4 took an additional step towards abstraction by considering the comparison of *multiple* topic models, as influenced by the tasks for which we use single models. While these sorts of high level inquiries are perhaps the most pertinent to answering researchers' questions about their data, they are dependent on a much lower level task: being able to look at a representation of a topic and decide what it is *about*. The second part of this dissertation will focus on a collection of experiments designed to validate some of the encodings used for this lower level task.

In this chapter, I will discuss a task that I will define as **gist-forming**. Gist-forming is the act of a user building a general sense of the semantic content of the words contained within a topic and of grasping the overarching concept or idea that connects them all—if such a connection exists. In many models, some topics are just the result of a statistical coincidence without any meaningful association of the words. Consequently, an important task related to gist-forming is that of **topic evaluation**. When building their gist, a user cannot just be concerned with what the topic is about, but also how much its words actually go together, how meaningful it is, and whether or not they can *trust* it to provide insight.

There are a variety of factors that may affect a user's ability to form a gist from a topic. These include: the visual encoding used, which can range from lists of words, to bar charts, to word clouds; the number of words included in the representation of the topic; and the semantic quality or cohesiveness of the topic itself. We conducted a set of crowdsourced experiments that used concrete subtasks to explore the ways in which these factors influence the abstract tasks of gist-forming and topic evaluation. We found that some factors matter more than others. In particular, though we hypothesized that visual encoding would have the greatest effect, performance was remarkably resistant to changes in encoding. Far greater were the effects of topic quality. In the following sections, we lay out the experiments we designed to discover this effect, and describe their implications for architects of topic model visualizations.

The primary contributions of this chapter to the field are:

- An articulation of the task of gist-finding for future investigation.

- An experimental mechanism for assessing it, combining the subtasks of **topic naming** and **word matching**.
- An evaluation of the factors of visual encoding, number of words, noise, and topic quality for their influence on this task.

Work presented in this chapter was originally published as Alexander and Gleicher (2016).

## 5.1 Related work

As described in Section 2.3, there are a few common methods for conveying topics being used in topic modeling visualization systems, most notably word lists, bar charts, and word clouds. Word lists and word clouds are the most popular, and in some ways the most visually disparate, so we decided to primarily focus on them for this work.

As also mentioned in Chapter 2, critiques of these encodings have largely focused on word clouds, though few of them provide much empirical backing. The study that perhaps provides the most evidence against the use of word clouds is in Rivadeneira et al. (2007). The authors consider tasks including search, browsing, impression formation, and word recognition (Rivadeneira et al., 2007). However, though their description of impression formation is similar to gist-forming, their experiments do not match our goals for topic representations. They determine that search and browsing are easier with a simple sorted list, which is understandable given the organization offered by alphabetical ordering—a finding also supported by Halvey et al. (Halvey and Keane, 2007). They focus on the task of recognition—recall of specific words seen in the cloud, and the ability to distinguish from similar but absent words—which is the opposite goal of a generalizable gist (described more in Section 5.2). While they penalize participants for identifying words that do not appear in the word cloud but are related, topic gists *need* to extend to words other than those explicitly contained in the visualization. In this way, what Rivadeneira et al. identified as a limitation of word clouds for their tasks could act as a *strength* for the task of gist-forming. The authors of this study also put explicit time limits of 20 seconds on all tasks, negating the possibility that some encodings might encourage longer engagement through greater aesthetic appeal.

## 5.2 General experimental design

Having a good "gist" of a topic is an abstract concept, making it difficult to quantitatively measure. It is important that the gist derived by a user is generalizable beyond just the specific words that they see in the representation. This is because given the size of most topics, any representation will necessarily display only a subset of the words that the topic contains. As mentioned in Section 5.1, generalizability is almost the opposite of the "recognition" goal used in Rivadeneira et al. (2007), which penalized participants for recalling words that were similar to the words shown, but not actually present. It is also important that the user be able to form this general sense *quickly*, though this issue is more subtle. On one hand, being able to form an accurate idea at a quick glance is valuable, but so too might be a visualization that encourages longer linger time and engagement (Hearst and Rosner, 2008).

To evaluate the abstract task of gist-forming, we developed an experimental procedure that combines two concrete tasks—**topic naming** and **word matching**—with measures of participant confidence.

**Topic naming** In the topic naming task, participants are presented with a representation of a topic and asked to provide a name that captures the essence of the topic as closely as possible. This is meant to induce the process of gist-forming, as a user cannot create a properly descriptive name without first coming up with a cohesive idea of what the topic is about. This task was inspired by watching our domain collaborators build an understanding of a model through the process of exploring its words and documents and creating names for each topic. Given the subtlety of meaning that can be contained in a name, however, *correctness* of these names is often subjective and difficult to accurately measure.

**Word matching** Trying to evaluate topic names for some concept of correctness is difficult, given the wide variety of valid names that may fit. For this reason, we pair the naming task with a word matching task to evaluate the robustness of participants' gists, similar to the word intrusion task introduced in Chang et al. (2009). In it, we hold out a small number of high-ranking words from each topic that is presented. We then show participants these words, mixed with a roughly equal number of highly ranked words from unrelated topics, and for each word ask them to decide if it could have come from the represented topic. This allows us to compute objective accuracy measurements to assess how well a participant's

concept of a topic can be generalized to other words that they might see in the context of that topic.

**Confidence** To measure how hard these tasks seemed for participants, as well as the amount of trust that they put into their own gists, we had them report their confidence in their answers. These took the form of scores from 1 to 7 for both their confidence that their name fully captured the nature of the topic and their confidence in whether or not each word belonged (see Figure 5.1). It is worth noting that we are not always looking for complete confidence. Due to the probabilistic nature of topic models, there are times when users should be cautious in the conclusions they draw, especially for lower quality topics.

In a pretesting phase, we asked for both user confidence as well as their opinion of the topic's quality. However, these two questions were strongly correlated enough to seem redundant, and we wanted to avoid having to explain to participants precisely what topic quality means, which is why we only asked for their confidence in these experiments.

In addition to measuring word matching accuracy and these two kinds of confidence (confidence in topic name and confidence in having matched the correct words), we also measured the time that participants spent with each topic representation. Though we gave them unlimited time to complete each question, we wanted to see whether or not linger time differed across factors or correlated with any of our other measures.

Topic contents and representations differed across experiments, as will be described in Section 5.3 and Section 5.4, but all used the same basic experimental setup. After giving consent, participants were shown a brief tutorial explaining the experimental task and the different encodings they might see. They were then presented with a succession of stimuli of the form shown in Figure 5.1 (in random order). On a single web page, they were asked to look at the topic representation that was shown, input a name for the topic and a number indicating their confidence in that name (from 1 to 7). They then were presented with a selection of words not contained in the representation and asked if they seemed to belong with the topic. This binary choice ("yes" or "no") was accompanied with a confidence score (also from 1 to 7). After a participant had completed all of their allotted questions, they were asked to input demographic data and any comments.

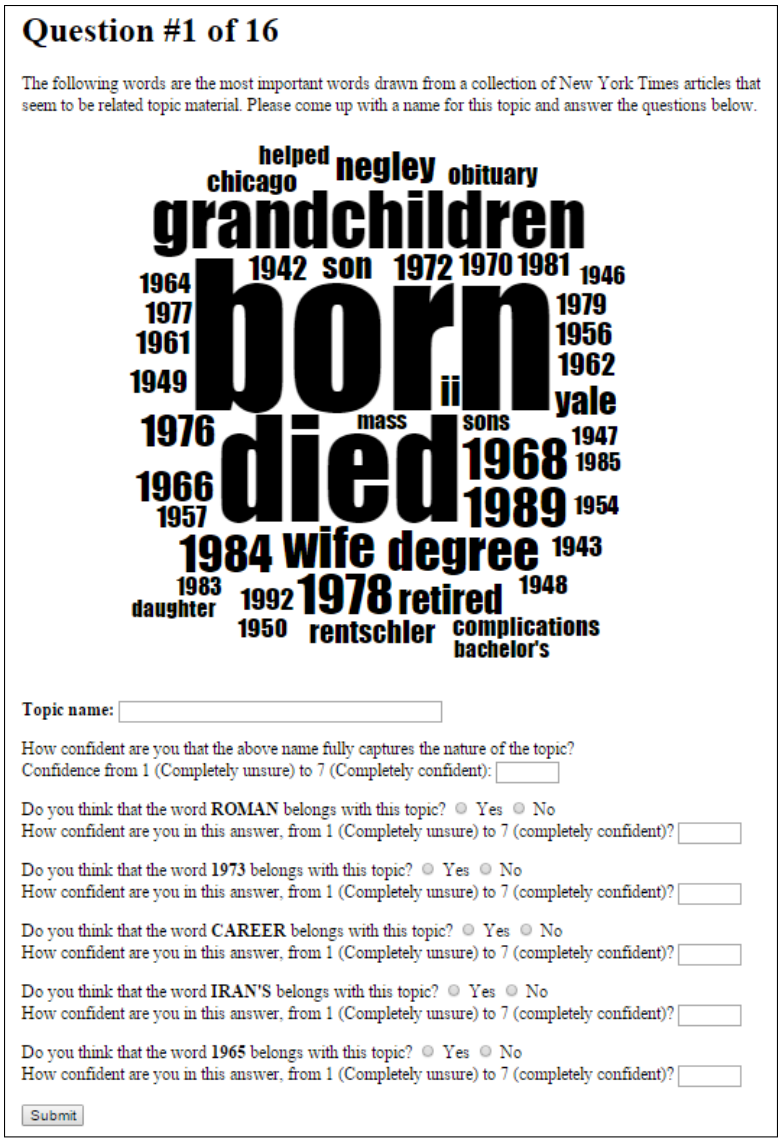



Figure 5.1: This is an example of a stimulus that might have been presented to a participant. This particular representation is in the word cloud category.

### 5.2.1 Stimuli

The topics we used for these experiments were drawn from a model built with 100 topics on a collection of New York Times articles from 2006 (Sandhaus, 2008). The model was built with the Latent Dirichlet Allocation (LDA) algorithm (Blei et al., 2003) as implemented in the Gensim Python library (Řehůřek and Sojka, 2010). The topic representations were created with the D3 visualization library for Javascript (Bostock et al., 2011) and Jason Davies' d3-cloud extension (Davies, 2015), employing an Archimedean spiral technique that places the largest words towards the center of the visualization.

### 5.2.2 Participants

Over the course of four experiments, we recruited 111 participants using Amazon's Mechanical Turk framework, specifically restricted to native English speakers residing in North America with at least a 95% approval rating. These participants ranged in age from 19 to 65 (with a mean of 33) and were made up of 64 males and 47 females. We paid participants $2.00 for their time.

## 5.3 Comparing visual encodings

The first factor impacting gist-forming that we evaluated was the designer's choice of visual encoding. In particular, we compared the two encodings most commonly used in practice: word lists and word clouds. Word lists are a subset of the words in a topic—the most frequent ones—displayed in descending order by frequency (see Figure 5.2 for examples of this technique). Word clouds are pictures displaying weighted lists of words, with weight—in this case, frequency within the topic—encoded using font-size (see Figure 5.1 for an example). We decided upon these two encodings both for their ubiquity in the literature and online, as well as for their distinct difference in appearance. We felt this would make them most likely to expose differences in user performance.

| "Good" topics | | | "Mediocre" topics | | |
|---|---|---|---|---|---|
| air | bar | oil | island | images | report |
| force | beer | energy | long | years | department |
| side | drink | power | cat | work | agency |
| plane | ice | environmental | sound | photographs | officials |
| crash | cocktail | plant | animal | history | office |
| aircraft | made | gas | shore | early | investigation |
| safety | back | plants | bear | american | information |
| bags | glass | water | animals | ago | federal |
| pounds | coffee | fuel | deer | modern | government |
| vehicle | drinking | natural | people | native | general |

Figure 5.2: Examples of "good" topics and "mediocre" topics from our model built on New York Times articles.

Word clouds are an often polarizing visualization technique (Harris, 2011). While many of the arguments against them are more about how they are used than the encoding itself, it has been empirically shown that word clouds are poor at helping users do tasks like recall or searching for a particular word (Rivadeneira

et al., 2007). However, we hypothesized that they may be particularly suited to the task of gist-forming. They fit a large amount of data (which is to say, many words) into a compact space (Meeks, 2012). With proper layouts and sizing, they can quickly direct the user's eye to the most important words of the visualization (Lohmann et al., 2009). Their aesthetics may increase engagement and time spent with the visualization (van der Geest and van Dongelen, 2009). Finally, their popularity mean that most users are already equipped to interpret them.

Given these strengths, we hypothesized that:

- User accuracy would be at least as good when using word clouds as when using word lists.
- Users would take longer with word clouds (i.e., longer linger time).
- Users would *prefer* word clouds to word lists.

### 5.3.1 Experiment 1A: Good topics

For our stimuli, we hand-selected 16 topics from our New York Times model (see Section 5.2.1) that seemed to be highly cohesive, so as to avoid floor effects. In addition to our subjective impressions, we also confirmed that these topics scored highly on the Uniform Distribution ranking and Vacuous Semantic Distribution ranking proposed in (AlSumait et al., 2009). Using a within-subjects design, we presented each participant with 16 stimuli—8 word clouds and 8 word lists—each containing the top 50 most frequent words from their respective topics. The order of the stimuli was randomized, as were which representations were paired with which topics.

We recruited 23 participants (13 male, 10 female) on Amazon's Mechanical Turk with ages ranging from 19 to 47 (with an average of 31).

#### Results

We ran a series of two-way analyses of variance (ANOVAs) to look for effects of representation and word ranking on the measures of accuracy, word confidence, name confidence, and time taken. We saw no effects of representation on accuracy ($F(1, 154) = 0.13$, $p = 0.72$), name confidence ($F(1, 22) = 1.23$, $p = 0.28$), or time taken ($F(1, 22) = 3.26$, $p = 0.08$). We did see a significant effect of representation on the user's confidence for their individual word decisions ($F(1, 154) = 5.62$, $p = 0.02$), but the effect size was small (a difference in means of .19 on an integer

scale from 1 to 7—see Figure 5.4). Accuracy across the two representations was exceptionally close: when presented with a word cloud representation, participants correctly identified associated words at a rate of 0.866 as compared to a rate of 0.87 when presented with word lists. Despite resulting in such similar performance, nearly two thirds of the participants (14 of 22) expressed a preference for word clouds over word lists, generally citing reasons such as they were "easier to read."

We did see a significant effect of a word's ranking within the topic on both the participants' accuracy at correctly matching it ($F(3, 154) = 7.97$, $p < 0.0001$), as well as their confidence in said matches ($F(3, 154) = 48.17$, $p < 0.0001$). Accuracy and confidence went down the further down the topic's ranking a selected word was drawn from.

#### Discussion

It seems as though for topics as good as the ones selected for this experiment, the difference in visual representation is too small to matter. Though we hypothesized that word clouds would have at least as high performance as word lists, such consistency across all of our measures is surprising. These results suggest the question of whether participants are deriving the same interpretations across representations or if the topics are so good (having been selected for their coherence) that we are seeing an accuracy ceiling regardless of the representations' differences.

The effect of a word's ranking within a topic is encouraging to see, as it reinforces the use of such ranking schemes for creating topic representations (see Figure 5.6). Still, without having seen an explicit difference across conditions, more experiments were needed to ensure that the experimental mechanism was sufficiently sensitive to find differences when they exist.

### 5.3.2 Experiment 1B: Mediocre topics

To be sure that the similar performance we saw across visual encodings in Section 5.3.1 was not simply a factor of having picked the best topics possible, we ran a second experiment with a set of lower quality topics. While topics from the first experiment were selected to be as coherent as possible, these topics were selected to be "mediocre," in that they still seemed to show some level of semantic cohesion (i.e., they were not junk topics) but the connection between the words was harder to grasp. Though this was a subjective selection, they were also ensured to be topics that scored lower using objective topic rankings (AlSumait et al., 2009).

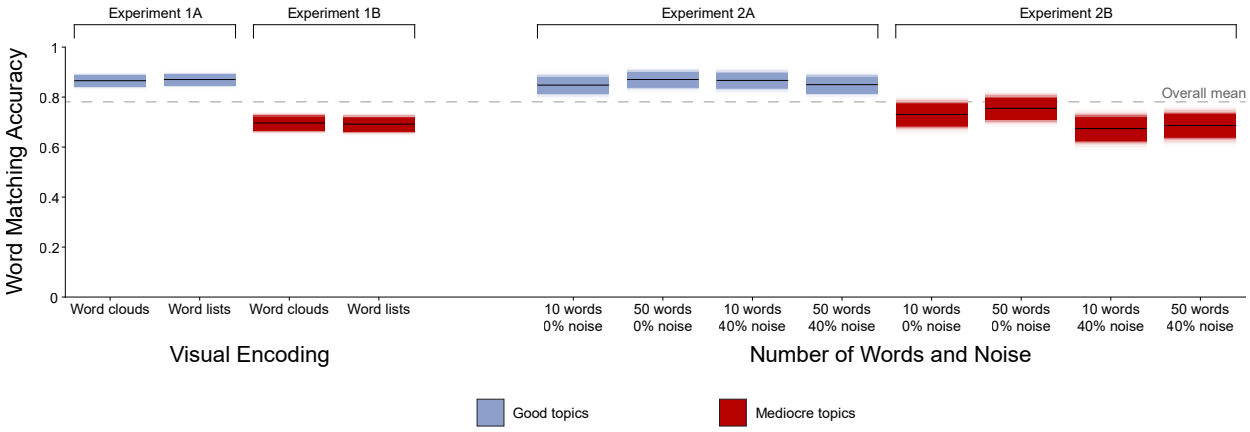



Figure 5.3: The effects of representation features on word matching accuracy, as gradient plots (Correll and Gleicher, 2014). Fully opaque colored regions represent a 95% confidence interval and an alpha gradient extends to the 100% confidence interval. We saw no significant effects of visual encoding or number of words. There was no effect of noise with good quality topics, and only a small effect with lower quality topics However, with the data combined as described in Section 5.3.2 and Section 5.4.2, we did see significant differences between topics of good and mediocre quality.

For this experiment, we used a within-subjects design identical to that in Section 5.3.1, with each participant seeing 8 word list stimuli and 8 word cloud stimuli, each containing a topic's top 50 words. However, we substituted the original 16 "good" topics for these new 16 "mediocre" topics.

We recruited 28 participants (19 male, 9 female) with ages ranging from 21 to 65 (with a mean of 34).

### Results

We ran a series of two-way ANOVAs to look for effects of representation and word ranking on accuracy, confidence, and time taken. There was once again no significant effect of representation type on accuracy ($F(1, 189) = 0.09$, $p = 0.76$), name confidence ($F(1, 27) = 1.85$, $p = 0.19$), or time taken ($F(1, 27) = 3.98$, $p = 0.056$). There was also no significant effect on word matching confidence ($F(1, 189) = 0.12$, $p = 0.73$). However, the overall accuracy with the new topics was lower than that measured in the first experiment, with a mean of 0.694 as compared to 0.868. A word's ranking was a significant factor in participant's confidence in their word matching ($F(3, 189) = 30.53$, $p < 0.0001$), though not in their accuracy ($F(3, 189) = 1.58$, $p = 0.20$). Participants favored word clouds over word lists at an even higher rate than before (21 of 28).

**Discussion**

Where we might have expected the lower quality topics to expose differences between the representations, performance remained steady across the two conditions. This seems to indicate that users' gist-forming abilities are, in fact, resistant to this shift in visual representation. However, the overall change in accuracy indicates a dramatic difference between topics of good quality and topics of mediocre quality.

This difference becomes particularly apparent when we look at the data from the two experiments together. It is worth noting that this was a sequence of two experiments, and not a proper single between-subjects experiment, as we ran the two experiments on different days. As the participant pool of Mechanical Turkers can vary, this is a potential source of variance that is unaccounted for when making this comparison, although the experiments were run at similar times each day. Furthermore, our measurements for these experiments were consistent with successive experiments as described in Sections 5.4.1 and 5.4.2.

We ran a series of two-way ANOVAs with the combined data to look for effects of representation and topic quality. We saw main effects of topic quality on accuracy ($F(1, 49) = 49.37$, $p < 0.0001$), word confidence ($F(1, 49) = 10.85$, $p = 0.002$), and name confidence ($F(1, 49) = 19.14$, $p < 0.0001$), each of which go down for topics of lower quality. There were no other main or interaction effects.

## 5.4 Number of words and noise

In our next set of experiments, we evaluated two different factors for their effects on gist-forming. The first was the number of words that the user is shown from the topic. In practice, this number can range anywhere from two or three to hundreds. On one hand, seeing more words would seem to be beneficial, as it gives the user more data to form their gist. This seems especially important given our observation in Section 5.3 that lower ranked words are harder to match with their topic. On the other hand, more words could be overwhelming to the user.

In addition to number of words, we also examined the effect of injecting noise into the topic words. For this factor, we replaced words in a topic representation with "noise words" that did not appear in that topic. We hypothesized that this mechanism might be a way of simulating topics of poorer quality in a way that we could quantifiably measure as the percentage of each topic made up of noise. When choosing noise words for a particular topic, we selected words that had

no significant probability in the topic's distribution, but did appear at the same position in a *different* topic in the model. This ensured that while the noise words shared no relation with the topic at hand, they did not stand out as completely obscure from the rest of the corpus.

For these factors, we hypothesized that:

- Providing more words would improve participant accuracy (though possibly hurt confidence).
- Introducing noise would decrease both participant accuracy and confidence.
- The improved accuracy with more words would be *more* pronounced with noisy topics.

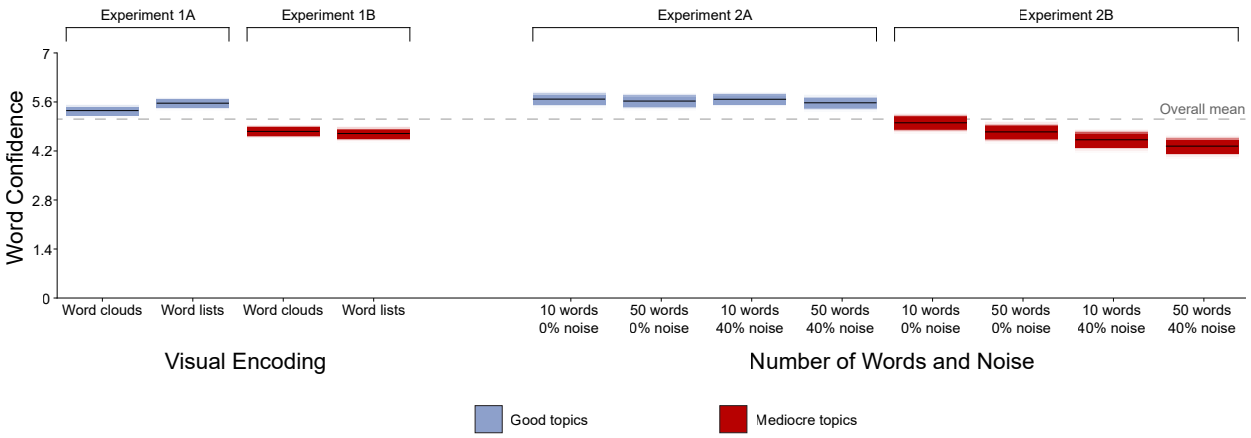



Figure 5.4: The effects of representation features on word matching confidence, as gradient plots (Correll and Gleicher, 2014). Fully opaque colored regions represent a 95% confidence interval and an alpha gradient extends to the 100% confidence interval. We see significant effects of noise and extra words in Experiment 2B (see Section 5.4.2), as well as a significant effect of topic quality with combined data as described in Section 5.3.2 and Section 5.4.2.

### 5.4.1 Experiment 2A: Good topics

For the first experiment exploring these factors, we used the same high-quality topics as in Section 5.3.1. We created a within-subjects design to look for any main or interaction effects between the two factors. We used two levels for the number of words factor (10 words and 50 words) and two levels for the noise factor (0% noise and 40% noise). Each participant again saw 16 stimuli, 4 from each combination of levels. All stimuli used the word list encoding. We collected

responses from 20 participants (11 male, 9 female) with ages ranging from 23 to 48 (with a mean of 33).

**Results**

We ran a series of two-way ANOVAs to look for effects of noise, number of words, and word ranking. The number of words factor exhibited no main effects on accuracy ($F(1, 284) = 0.63$, $p = 0.43$), word confidence ($F(1, 284) = 0.06$, $p = 0.81$), or name confidence ($F(1, 57) = 0.18$, $p = 0.68$). After excluding outliers that appear to have been instances of the participant leaving the computer for extended periods of time, participants did spend significantly longer on stimuli with 50 words ($F(1, 57) = 4.17$, $p = 0.04$), but this is to be expected given the longer time it would take to read.

While the presence of noise seemed to decrease participants' confidence in their topic names ($F(1, 57) = 45.56$, $p < 0.0001$), it had no discernible effect on either their accuracy ($F(1, 284) = 0.56$, $p = 0.45$) or their confidence when asked whether or not new words went with the topic ($F(1, 284) = 0.0001$, $p = 0.99$).

Once again, a new word's ranking within the topic had a significant effect on participant's accuracy ($F(3, 284) = 3.42$, $p = 0.02$) and confidence ($F(3, 284) = 21.85$, $p < 0.0001$) when matching it to its topic.

**Discussion**

We were surprised not to see an effect of the number of words included on either accuracy or confidence. The difference in magnitude from 10 to 50 words is drastic, with the latter group receiving five times as much information to work with as the former. The longer times spent on the questions with more words seem to indicate that participants were *looking at* the extra words, and yet the extra data offered no benefit for the word matching task.

The presence of noise words within the stimuli seemed to create a *perception* of difficulty without actually affecting performance in the word matching task. This is surprising, as we had expected introducing noise to be a way of artificially making the task harder, but participants appeared to be fully adept at *seeing through* the noise.

It is interesting to note that participants' overall accuracy (0.859) nearly matched that of the first visual encodings experiment that used the same "good"

topics (0.868), further reinforcing the resistance of the gist-forming task to changes in presentation.

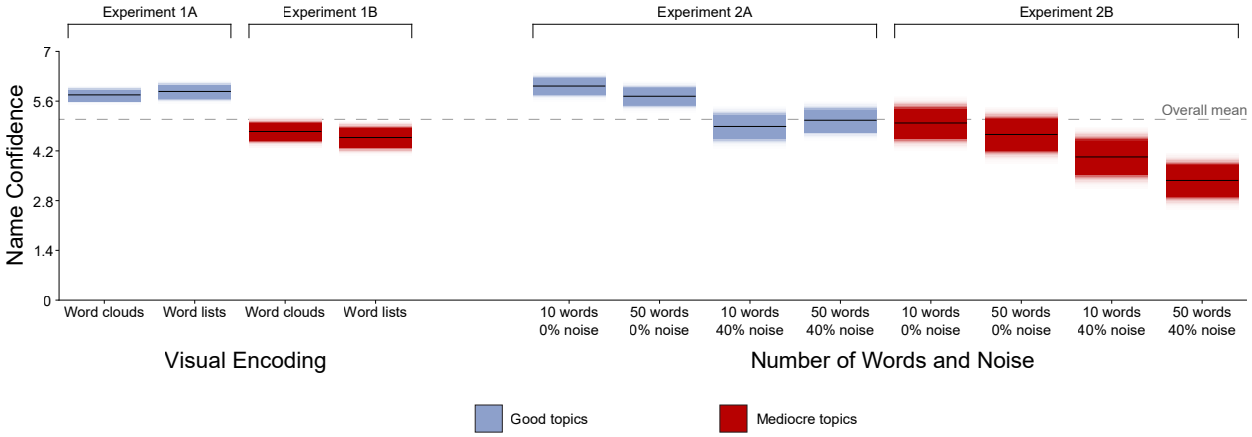



Figure 5.5: The effects of representation features on topic name confidence, as gradient plots (Correll and Gleicher, 2014). Fully opaque colored regions represent a 95% confidence interval and an alpha gradient extends to the 100% confidence interval. While there was no effect of visual encoding, noise resulted in significantly lower confidence, as did more words with mediocre topics. After combining the data from experiments together as described in Section 5.3.2 and Section 5.4.2, topic quality showed a significant effect on confidence, as well.

### 5.4.2 Experiment 2B: Mediocre topics

As before, we wanted to ensure that the consistency in accuracy we observed was not the result of ceiling effects associated with the high-quality topics. We ran an experiment using the same experimental design but switching out the high-quality topics for "mediocre" ones. This was again a within-subjects design, presenting each subject with 16 word list stimuli, 4 of each combination of noise levels (0% and 40%) and number of words (10 and 50). We recruited 38 participants (20 male, 18 female) with ages ranging from 23 to 60 (with a mean of 34).

#### Results

We ran a series of two-way ANOVAs to look for effects of number of words, noise, and word ranking. Once again, the number of words factor showed no significant effect on accuracy ($F(1, 238) = 2.18$, $p = 0.14$) or their confidence in their word matches ($F(1, 238) = 1.48$, $p = 0.23$). We did see a significant (though small) effect indicating that participants' confidence in their names ($F(1, 48) = 6.32$, $p = 0.02$) dropped in the 50-words condition (see Figure 5.5).

Introducing noise to the stimuli once again lowered participants' confidence in their names ($F(1, 48) = 32.94$, $p < 0.0001$) and in their word matches ($F(1, 238) = 30.45$, $p < 0.0001$). We also saw an effect of noise on accuracy that was not present with good topics, in which accuracy was slightly lower in the noisy case ($F(1, 238) = 4.68$, $p = 0.03$). However, the size of this effect was small ($M_{0\%} = 0.71$, $SD_{0\%} = 0.22$, $M_{40\%} = 0.65$, $SD_{40\%} = 0.23$).

No interaction effects between noise and number of words were observed. There were no effects to be observed on time taken to answer each question.

As in the previous experiments, a word's ranking within the topic had a significant effect on the participant's ability to match it to the representation ($F(3, 238) = 6.44$, $p = 0.0003$) and their confidence in said match ($F(3, 238) = 8.59$, $p < 0.0001$).

**Discussion**

Once again, our hypothesis for improved performance with more words was not substantiated. Confidence with more words actually went down—possibly indicating that participants were overwhelmed by the extra information (see Figures 5.4 and 5.5). Noise once again introduced uncertainty in the participants' responses without negatively affecting their word matching accuracy.

The overall accuracy for the word matching task was 0.711, down from 0.859 in the first experiment looking at these factors. Looking at the combined data from these two experiments reinforces this trend (though it must be done with the same caveats as described in Section 5.3.2). With the combined data, we ran a series of two-way ANOVAs looking for the effects of topic quality with number of words and noise. Upon doing this, we are able to see a main effect of topic quality on accuracy ($F(1, 35) = 86.58$, $p < 0.0001$), word confidence ($F(1, 35) = 16.14$, $p = 0.0003$), and name confidence ($F(1, 35) = 16.91$, $p = 0.0002$), each of which go down with the worse topics.

In these experiments, we see again that topic quality has a dramatic effect on both accuracy and confidence. We also find that noise turns out not to be a good way of simulating poor topics. Introducing noise to good topics did not result in a decrease in accuracy, while replacing good topics with mediocre topics resulted in a substantial decrease in accuracy. However, it is still very interesting that participants' gists were able to withstand that level of manipulation.

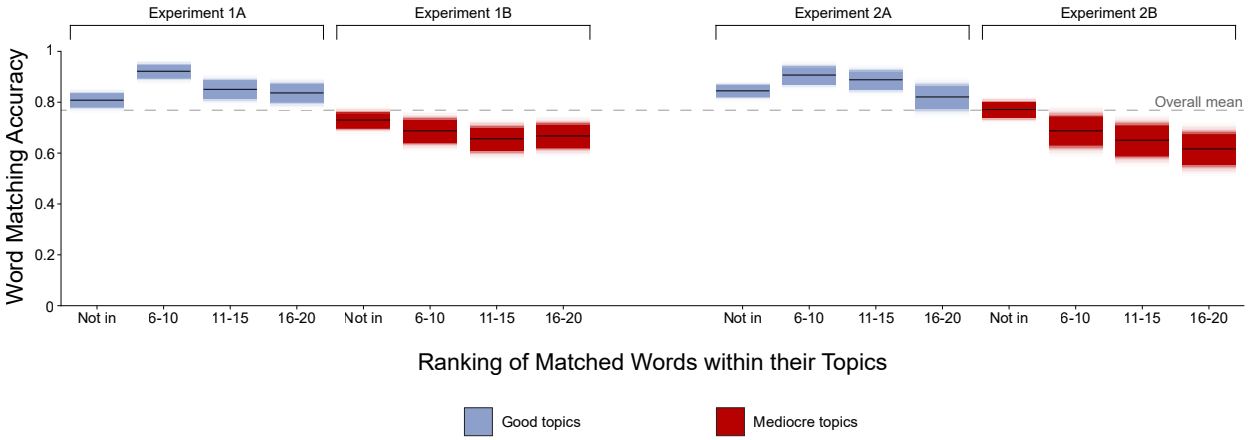



Figure 5.6: The effects of word ranking on accuracy, as gradient plots (Correll and Gleicher, 2014). Fully opaque colored regions represent a 95% confidence interval and an alpha gradient extends to the 100% confidence interval. When selecting words to be matched with the topic representation, we drew from three different sections of the topic rankings: the 6-10 ranked words, the 11-15 ranked words, and the 16-20 ranked words. Here, we plot participant accuracy by these word groups (along with words that were drawn from outside of the topic).

## 5.5 Full discussion

We are able to draw a number of takeaways from this exploration of gist-forming that are relevant to designers of topic modeling tools and visualizations. Counter to our expectations, gist-forming seems to be quite robust to changes in the visual encoding used to convey topics. The robustness of user performance across encoding, combined with participants' preference for word clouds over word lists, may indicate that word clouds are suitable to use for topic interpretation tasks, despite their documented poor performance in helping users with other tasks such as search and recall. It is possible that other encodings (e.g., bar charts) may differ in ways not captured by the pairwise experiments described here, but we believe word clouds and word lists are representative of the literature.

Gist-forming also appears to be resistant to changes in the number of words shown, which did not affect accuracy in either Section 5.4.1 or Section 5.4.2 and showed only a minor effect on confidence in Section 5.4.2. The drop in confidence seen with more words on worse topics may suggest that number of words can be used by designers as a method of tempering the tendency in some users to make overly broad generalizations about what topic trends may mean.

Similarly, the drop in user confidence for both mediocre and noisy topics is beneficial for the task of topic evaluation. These lower quality topics are instances

when one would *want* user confidence to go down—for users to form their interpretations with a grain of salt rather than making sweeping claims based on tenuous connections. As users seem to be able to differentiate between topics of different quality, designers may be able to leave the task of topic evaluation largely in their hands.

Finally, it is clear that the factor with the greatest effect on the gist-forming task is topic quality. While creating good visualizations can help users achieve new insights, this finding reinforces the need to help them arrive at good models. Tools that incorporate users into the *training* process are crucial to this effort.

The takeaways of this chapter increased my own trust in word clouds as a means of topic conveyance, and led me to introduce word clouds into my own topic model exploration (as described in Section 3.4). However, this specific focus on gist-forming still leaves open the question of such an encoding's use more generally in visualization. In the next chapter, I will discuss work looking into a more specific set of tasks for font size encodings, seeking to answer the question "Can users correctly interpret the values being presented?"

# 6 ENCODING DATA WITH FONT SIZE

*Now I perceive that she hath made compare*
*Between our statures; she hath urged her height,*
*And with her personage, her tall personage,*
*Her height, forsooth, she hath prevailed with him.*

— Hermia, *A Midsummer Night's Dream*

The last chapter looked at a comparison of different methods of conveying individual topics to help researchers form a gist of them within a topic model. In this evaluation, user performance when using word clouds was at least as good as when using word lists. While this is encouraging regarding the use of word clouds in these specific situations, it leaves many open questions about their use in other situations. Notably, it is not clear from the studies in Chapter 5 to what degree (if any) the actual *data encoding* is creating a benefit for users. Given the comparison being made was against word lists that only encode rank (rather than value), it is possible that users were not considering the values being conveyed through font size at all. This chapter will seek to address this gap in our knowledge in a way that will provide extensible knowledge about the use of font size as a data encoding in many situations.

Font size is a common method for encoding data in a variety of domains. The importance and impact of font size as an encoding go beyond visualizations of text collections, as words of varying sizes are embedded as labels in many other types of visualizations. Font size encodings can be seen in word cloud applications (as described in earlier chapters and seen in (Kuo et al., 2007; Trattner et al., 2014; Viégas et al., 2009)), cartographic labeling (Afzal et al., 2012; Skupin, 2004), and a number of different hierarchical visualization tools (Brath and Banissi, 2015; Wattenberg and Viégas, 2008).

It is important to understand how the visual features of words and letters can bias interpretation of font size, and by extension interpretation of the underlying data. There has been some question of how effective people are at judging font size encodings (Hearst and Rosner, 2008). Concerns about these encodings arise in part because of the various ways in which words vary with one another outside of font size. In particular, two words with the same font size can vary tremendously in their *shape*. Longer words with more letters take up more area on the screen. The glyphs for some letters are inherently taller or wider than others. Kerning and tracking can create diverse spacing between characters. Differences in font would exacerbate these problems, but even the same font is often rendered differently depending on the platform. Other potentially biasing factors include color, font weight, and a word's semantic meaning (Bateman et al., 2008; Kuo et al., 2007; Lohmann et al., 2009; Rivadeneira et al., 2007; Viégas et al., 2009).

In this chapter, I describe evaluations of the degree to which a word's shape can affect impressions of its font size. I present the results from a series of crowdsourced experiments in which participants were asked to judge font size

within word cloud visualizations. In each experiment, we varied the words along one of the axes described above. We found that, in general, performance was quite high—surprisingly so. There were conditions in which participants' perception of font size was biased. In particular, in cases where perception of some physical attribute of the word, such as width, *disagreed* with the perception of font size, accuracy dropped for many participants.

Fortunately, this effect can be corrected for. I describe a proof-of-concept method for debiasing font size encodings that uses colored tags sized proportionally to the data. Our work empirically shows that our debiasing efforts improve performance even in the most pathological cases.

The main contributions contained within this chapter are:

- An evaluation of user accuracy when making comparative judgments of font size encoding within a visualization, indicating that users may be better at making such judgments than conventional wisdom would suggest.
- A description of situations in which these judgments can be biased by attributes of the words being shown.
- A proof-of-concept method for debiasing visualizations in these situations using padded bounding boxes.

Work in this chapter is currently in submission to the IEEE Transactions on Visualization and Computer Graphics.


Figure 6.1: To test whether attributes of words can affect perception of their font size, we highlighted words within word clouds and asked participants to choose the larger font. On the left, "zoo" has the larger font, but the length of "moreover" can bias participants toward choosing it as larger. On the right, "source" has the larger font, but the taller ascending and descending parts of "begged" can bias participants toward choosing it as larger.

## 6.1 Related work

I describe some of the uses of font size encodings as specific to topic modeling visualization in Chapter 2. Here, I will discuss literature related to the use of font size more broadly.

Font size has been used to encode data across a number of visualization types, and to support a variety of tasks. Investigations of font size encoding have been largely focused on word clouds and their overall effectiveness, whereas the work in this chapter focuses on the perceptual task of comparing word sizes under a variety of real-world conditions.

Word clouds (or tag clouds) are among the most familiar visualizations using font size encodings, as can be seen in tools like Wordle (Viégas et al., 2009) and Word Tree (Wattenberg and Viégas, 2008). Font size has also been used to encode data in cartographic visualizations, in typographic and knowledge maps. A typographic map represents streets using textual labels for street names while encoding spatial data such as traffic density, crime rate, or demographic data into the font size (Afzal et al., 2012; Maps, 2015). In contrast, Skupin uses font size to indicate semantic clustering, which allows users to zoom in and out of his knowledge maps semantically (Skupin, 2004).

A study by Bateman et al. investigates the visual influence of word cloud visual properties (font size, tag area, tag width, font weight, number of characters, color, intensity and number of pixels) for the task of selecting the 10 “most important tags” (Bateman et al., 2008). Participants were asked to find the most attention-grabbing word out of a word cloud. They report that the features exerting the greatest visual influence on word clouds were font size, font weight, saturation and color. However, the authors did not look at user ability to accurately read data encoded with these features.

A study by Lohmann et al. reports that words with larger font sizes attract more attention and are easier to find. However, none of these studies identify the magnitude of this effect for real-world use, or strategies for mitigating the biases (Lohmann et al., 2009). This knowledge is relevant because when encoding data into font size (Afzal et al., 2012; Nacenta et al., 2012; Skupin, 2004; Wattenberg and Viégas, 2008) there is expectation from designers that people can perceive the difference in size to correctly understand the encoded data.

## 6.2 Experimental task

There are many different documented tasks for which font size encodings have been used. These tasks include:

- **Gist-forming**: discerning the general meaning of a collection of words, considering their relative importance as coded by their font size (Chapter 5).
- **Summary comparison**: making sense of juxtaposed sets of words from different sources (Alper et al., 2011; Collins et al., 2009b).
- **Word search**: finding a particular word in a visualization (Bateman et al., 2008; Lohmann et al., 2009; Rivadeneira et al., 2007).
- **Retention**: being able to recall a word from a particular visualization, and to distinguish it from others (Rivadeneira et al., 2007).
- **Value reading**: reading a specific numerical value associated with text (Nacenta et al., 2012).
- **Order reading**: comparing words to determine relative value (Bateman et al., 2008; Rivadeneira et al., 2007).

It has been shown that font size encodings are not the proper design choice for a number of these tasks, most notably searching and retention, where simple ordering can be much more effective (Rivadeneira et al., 2007). In general, font size encodings are more frequently used for subjective, high-level tasks such as the gist-forming task described in Chapter 5. However, it is difficult to measure perceptual limitations with these tasks. For this study, we were not interested in measuring participants' cognitive ability to draw connections between groups of words, but rather in better understanding their *perceptual* abilities.

As such, in selecting a task for our experiments, we chose one that we believed would isolate the primitive *sub-task* of discerning information represented in font size. Specifically, we focused on a simple comparison task. We would highlight two words within a visualization containing words of different sizes and ask subjects to choose the one with the larger font size. The ability to make accurate relative judgments of represented data is a prerequisite for such tasks as gist-forming, summary comparison, order reading, and value reading. Therefore, though users in the wild are rarely faced with a single pairwise comparison, we

believed performance at this task would help us measure the ability to perform higher level tasks that rely on the same perceptual abilities.

There were other tasks that we considered, as well. One solution might have been to ask participants to make an absolute judgment of font size (e.g., 1.5mm), or to compare to a memorized baseline size (e.g., bigger than baseline). Although such tasks are simple, their detachment from the context of real-world tasks might have lead to idiosyncratic strategies, such as focusing attention on the height of a single letter instead making a holistic judgment about a whole word. At the other extreme, another solution might have been to ask which word in an entire cloud has the biggest font, while systematically manipulating the distribution of font sizes within that cloud. However, this task presents many degrees of freedom that make precise measurement more difficult. For example, it is not clear whether we should measure precision as the difference between the biggest font versus the next biggest, of versus the algebraic or geometric mean of the distribution, or versus some other property of the distribution (Haberman and Whitney, 2012; Ross and Burr, 2010; Szafir et al., 2016). We chose to use the pairwise comparison task in most of our experiments for the greater control it offered us. After having explored perceptual biases in this task, however, we still wanted to be sure that what we had found was extensible to more real-world situations, and so we ran a set of experiments using the pick-the-biggest-word task, which showed similar results (see Section 6.6).

## 6.3 General experimental design

As discussed in Section 6.2, we focused on *comparative* judgments of size rather than exact ones. In particular, we focused on the use of word clouds. Not only are these one of the most common mediums for font size encodings, but they also present a challenging context for reading values, given the dense proximity of distracting words and the frequent lack of alignment to any shared baseline for any pair of words.

### 6.3.1 Task Setup and Measures

Participants were first given instructions on the task, and read a tutorial indicating the difference between a word's font size and the area it took up on the screen. Participants were instructed to complete the tasks as accurately as possible.

Across multiple experiments, we gave participants the following task with different stimuli: Upon being shown a word cloud in which two words were highlighted using a darker gray, participants were asked to click on the highlighted word that had been given the larger font size. We were sure to fully explain the distinction between font size and the general footprint of a word on the screen. While others have observed instances of users misinterpreting the *meaning* of font size encodings (Viégas et al., 2009), we were concerned primarily with perceptual abilities, and so did not want there to be any confusion for participants.

For each task, we recorded which word the participant clicked, as well as the time it took. We measured time only to test for fatigue effects (were tasks getting slower over time, or was performance decreasing)—our primary measure was accuracy. Upon clicking a word, the participant was immediately presented with the next trial.

### 6.3.2 Factor Agreement

In each experiment, we tested a potentially biasing word factor to see if it affected the perception of font size. These factors were features of the words that vary based on the *contents of the words themselves*, such as word length, rather than attributes of the font that could feasibly be controlled across the entire visualization. To check for bias of a factor, we employed a method we have called **factor agreement**.

Factor agreement indicates whether the difference in the factor in question *reinforces* or *opposes* the difference in font size (see Figure 6.2). For example, if the word within a given pair with the larger font size also contains more letters, then we would say that word length **agrees** with font size. However, if the word with the larger font size contains fewer letters, we would say word length **disagrees** with font size. If both words are the same length, then the word length factor is **neutral**. It is not necessarily the case that any given factor's agreement or disagreement will affect a user's perception of font size, but if they do have an effect, we would expect user accuracy to decrease in situations of disagreement.

### 6.3.3 Stimuli

Stimuli for these experiments were all generated within a web browser. For early experiments, we created our own clouds using the D3 visualization library (Bostock et al., 2011). In later experiments, to create more realistic scenarios, we used

| | Factor agreement | | | | | |
|---|---|---|---|---|---|---|
| Factor | agree | | neutral | | disagree | |
| word length | hello<br>sam | bigger font,<br>longer word | hello<br>world | same length | hello<br>goodbye | bigger font,<br>shorter word |
| word height | help<br>corn | bigger font,<br>taller word | plot<br>flop | same<br>"raw height" | corn<br>help | bigger font,<br>shorter word |
| word width | joyful<br>letter | bigger font,<br>wider word | litter<br>fillet | same<br>"raw width" | little<br>hummed | bigger font,<br>narrower word |

Figure 6.2: In this figure, we show examples of the different conditions of **factor agreement** (see Section 6.3.2) for the three main factors of word shape that we tested: word length, word height, and word width. For height, we were concerned with the use of tall and short characters, rather than height differences resulting from font size. Similarly, for word width, our primary concern was not the *final* width of the word in the stimulus, but rather the *raw width*—its width before any changes in font size had been applied. While "litter" is wider than "fillet" in the above figure, they are the same width when written in the same font size.

jQCloud (Ongaro, 2014), a word cloud library that packs words more densely using a spiral layout. With the exception of Experiment HEIGHT3, in which we explicitly decided to test a sans serif font (see Table 6.1), we used Times New Roman for all of our stimuli.

The words used in each experiment were either English words or "pseudowords" (see Table 6.1). Pseudowords were strings of random characters that we created for greater control over the character glyphs being used, factoring out any semantic weight. Precise characteristics of these pseudowords varied between experiments (see Section 6.4). When building word clouds with English words, we drew from the Corpus of Contemporary American English (COCA) (Davies, 2011). We built a database that allowed us to query for words with specific attributes (e.g., length).

The two **target words** between which participants had to choose varied in their font sizes and attributes from experiment to experiment. They were also joined by 40 **distractor words** in each stimulus, whose sizes were distributed across a normal distribution. After some calibration through pilot studies, we kept the difference in font size between the two target words relatively small. Accuracy was high enough in these conditions that testing larger differences was deemed unnecessary.


Figure 6.3: For many of our experiments, we used word clouds that we built using the D3 visualization library (Bostock et al., 2011). These clouds dispersed words randomly throughout the two-dimensional space, restricted only by avoiding collisions with the borders and other words. Words were either drawn from the English words within COCA (Davies, 2011) or pseudowords created using random characters (as shown here).

One issue that came up during experimentation was how different browsers perform subpixel-rendering. For non-integer font sizes (e.g., 12.5px), modern browsers sometimes use different rendering methods that can result in participants with different machines viewing slightly different sizes. Browser difference would be a between-subjects factor, and so it should not affect within-subjects factors which account for the vast majority of those in our experiments. Additionally, the experiments we chose to report in the main body of the chapter all used integer-value font sizes. However, it is worth noting that some of the between-subjects effects described in Appendix A may be influenced by cross-browser differences.

### 6.3.4 Participants

Over 12 experiments, we recruited 301 participants using Amazon's Mechanical Turk framework, restricted to native English speakers residing in North America with at least a 95% approval rating. These participants ranged in age from 18 to 65 (with a mean of 33) and were made up of 172 males and 129 females. We paid participants either $1.00 or $2.00 for their time, depending on the number of stimuli with which we presented them (which varied from 56 to 150).

Within each session, we included "validation stimuli" with font size differences of a full 10 pixels. These validation stimuli were used as engagement checks to verify that participants had properly understood the instructions, and were not considered in further analysis.

## 6.4 Exploring biasing factors

Over the course of our explorations, we ran over a dozen experiments involving hundreds of participants on Amazon's Mechanical Turk. Rather than describe the results for every experiment in detail, the main results and takeaways from each experiment are organized into Tables 6.1 and 6.2. A subset of them will be discussed in greater depth in this section. The remaining experiments are described in full in Appendix A. The experiments are structured by the main factors that we tested for bias: word length, character height, and word width (shown in Figure 6.4).

### 6.4.1 Word Length

The first potentially biasing word attribute we tested was **word length**: the number of characters contained within a word. Longer words take up more space, and have a larger *area* than shorter words of the same font size, and even some shorter words with *larger* font sizes. We predicted that these differences in area could interfere with the ability to perceptually distinguish words by pure font size alone.

We ran four total experiments using word length as a test factor. In each one, we observed a significant effect in which participant accuracy went down when word length disagreed with font size. The details for these experiments can be found in Tables 6.1 and 6.2, as well as Appendix A. Two of the most important experiments are described here.

| Label | E/P | Effect of △ font size | Primary bias factor | Effect of factor agree- | Additional factor | Accuracy at min △ font size | | | Notes |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | agree | neutral | dis-agree | |
| len1 | P | ✓ | word length† | ✓ | - | 0.860 | 0.879 | 0.753 | Word length biases perception of font size |
| len2 | P | ✓ | word length† | ✓ | base font size‡ | 0.861 | 0.816 | 0.734 | We see a greater bias at larger base font (30px vs. 20px) |
| len3 | P | ✓ | word length† | ✓ | base font size† | 0.825 | 0.838 | 0.642 | Tested wider variety of baseline font sizes |
| len4 | E | ✓ | word length† | ✓ | - | 0.992 | 0.942 | 0.867 | Bias still present with English words and denser word clouds |
| height1 | P | ✓ | word height† | ✓ | - | 0.974 | 0.909 | 0.684 | Character heights bias perception of font size |
| height2 | P | ✓ | word height† | ✓ | - | 0.929 | 0.810 | 0.529 | Proportional difference in font size seems to matter more than absolute difference |
| height3 | P | ✓ | word height† | ✓ | - | 0.937 | 0.795 | 0.525 | Bias still present when word clouds use sans serif font |
| height4 | P | ✓ | word height† | ✓ | base font size† | 0.931 | 0.790 | 0.479 | We see a greater bias at larger base font (30px vs. 20px) |
| height5 | P | ✓ | word height† | ✓ | base font size‡ | 0.963 | 0.854 | 0.489 | Accuracy hits ceiling between 20-25% size difference |
| width1 | E | ✓ | word width† | ✓ | - | 0.975 | - | 0.909 | Bias present when length is held constant and width varies |
| width2 | E | ✗ | word length† | ✗ | - | 0.982 | - | 0.982 | No bias when width is held constant and length varies |
| box1 | E | ✓ | word width† | ✗ | - | 0.914 | 0.932 | 0.908 | No bias with corrected-width rectangular bounding boxes |
| big1 | P | ✓ | word length† | ✓ | number of near misses | 0.888 | 0.826 | 0.658 | Tested using "pick the biggest word" task |
| big2 | P | ✓ | word length† | ✓ | number of near misses | 0.811 | - | 0.562 | Tested wider variety of length differences |

† - within-subjects factor ‡ - between-subjects factor

Table 6.1: An overview of the experiments we ran for this study. Each experiment compared at least two factors: the difference in font size between the two target words, and a potentially biasing factor that was a feature of the words' shape. (Additional factors tested are described in Appendix A.) Here, we report the effects of these factors and the effect size of factor agreement at the smallest difference in font size tested (generally a 5% difference). Experiments with a white background are described in Sections 6.4 and 6.5, while those with a gray background are described in full in Appendix A. In column "E/P", "E" indicates that English words were used and "P" indicates that "pseudowords" were used (see Section 6.3.3).

### Experiment LEN1

For our first experiment on word length, we presented participants with word clouds of our own creation as described in Section 6.3.3 (see Figure 6.3). To afford greater control in stimulus generation, we used words of random characters, excluding characters with ascenders or descenders (e.g., "h" or "g"—see Figure 6.4) as well as characters of abnormal width (e.g., "w" or "i"). We enforced a minimum

| Experiment | N | Factors | Conditions | Analysis of Variance | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | W/B | df1 | df2 | F | p-value |
| len1 | 31 | Δ font size | w1: [20, 21, 22px], w2: [20, 22px] | W | 1 | 150 | 59.21 | < 0.0001 |
| | | word length agreement | w1: [5, 8 chars], w2: [5, 8 chars] | W | 2 | 150 | 14.91 | < 0.0001 |
| len2 | 39 | Δ font size | [5, 10, 15, 20%] | W | 3 | 418 | 58.96 | < 0.0001 |
| | | word length agreement | w1: [4, 7, 10 chars], w2: [4, 7, 10 chars] | W | 2 | 418 | 12.13 | < 0.0001 |
| | | base font size | [20, 30px] | B | 1 | 37 | 7.98 | 0.008 |
| len3 | 20 | Δ font size | [5, 10, 15, 20%] | W | 3 | 926 | 85.43 | < 0.0001 |
| | | word length agreement | w1: [5, 8 chars], w2: [5, 8 chars] | W | 2 | 926 | 31.60 | < 0.0001 |
| | | base font size | [20, 25, 30, 35px] | W | 3 | 926 | 8.57 | < 0.0001 |
| len4 | 20 | Δ font size | w1: [20px], w2: [21, 22, 23, 24px] | W | 3 | 269 | 7.84 | < 0.0001 |
| | | word length agreement | w1: [5, 8 chars], w2: [5, 8 chars] | W | 2 | 269 | 14.32 | < 0.0001 |
| height1 | 32 | Δ font size | w1: [20, 21, 22px], w2: [20, 22px] | W | 1 | 155 | 55.31 | < 0.0001 |
| | | word height agreement | w1: [tall, short], w2: [tall, short] | W | 2 | 155 | 71.22 | < 0.0001 |
| height2 | 20 | Δ font size | w1: [20, 22, 24px], w2: [21, 23px] | W | 5 | 323 | 45.88 | < 0.0001 |
| | | word height agreement | w1: [tall, short], w2: [tall, short] | W | 2 | 323 | 83.90 | < 0.0001 |
| height3 | 20 | Δ font size | w1: [20, 22, 24px], w2: [21, 23px] | W | 5 | 323 | 59.42 | < 0.0001 |
| | | word height agreement | w1: [tall, short], w2: [tall, short] | W | 2 | 323 | 36.10 | < 0.0001 |
| height4 | 20 | Δ font size | [5, 10, 15, 20%] | W | 3 | 448 | 59.81 | < 0.0001 |
| | | word height agreement | w1: [tall, short], w2: [tall, short] | W | 2 | 448 | 88.39 | < 0.0001 |
| | | base font size | [20, 30px] | W | 1 | 448 | 44.9 | < 0.0001 |
| height5 | 40 | Δ font size | [5, 10, 15, 20, 25%] | W | 4 | 546 | 94.39 | < 0.0001 |
| | | word height agreement | w1: [tall, short], w2: [tall, short] | W | 2 | 546 | 207.2 | < 0.0001 |
| | | base font size | [20, 30px] | B | 1 | 38 | 20.09 | < 0.0001 |
| width1 | 20 | Δ font size | w1: [20px], w2: [21, 22, 23, 24px] | W | 3 | 133 | 6.77 | 0.0003 |
| | | word width agreement | [+10px, -10px] | W | 1 | 133 | 11.33 | 0.001 |
| width2 | 19 | Δ font size | w1: [20px], w2: [21, 22, 23, 24px] | W | 3 | 126 | 1.47 | 0.23 |
| | | word length agreement | [+3 chars, -3 chars] | W | 1 | 126 | 0.00 | 1.00 |
| box1 | 20 | Δ font size | [5, 10, 15, 20%] | W | 3 | 209 | 10.88 | < 0.0001 |
| | | word width agreement | [-20px, 0px, +20px] | W | 2 | 209 | 0.52 | 0.60 |
| big1 | 19 | Δ font size | [5, 10, 15, 20%] | W | 3 | 414 | 5.82 | 0.0007 |
| | | word length agreement | target: [5, 8 chars], near misses: [5, 8 chars] | W | 2 | 414 | 10.10 | < 0.0001 |
| | | # near misses | [1, 4] | W | 1 | 414 | 33.66 | < 0.0001 |
| big2 | 19 | Δ font size | [5, 10, 15, 20%] | W | 3 | 846 | 3.02 | 0.03 |
| | | word length agreement | [-5, -3, -1, 1, 3, 5 chars] | W | 5 | 846 | 8.00 | < 0.0001 |
| | | # near misses | [1, 4] | W | 1 | 846 | 7.00 | 0.008 |

Table 6.2: An overview of the statistical tests run for this study. For each experiment, we show the number of participants (N), the factors and their levels (specifying conditions for both target words—w1 and w2—where appropriate), whether the factors were treated as within- or between-subjects factors, and analyses of variance for each. Other descriptive statistics can be seen in Table 6.1 and in Appendix A.

distance between the two highlighted words, and ensured that they shared no common horizontal or vertical baselines that would aid in comparison.

We tested two main factors: font size and word length. Both were examined using within-subject comparisons. Font size for the first target word was either 20px, 21px, or 22px, while font size for the second word was either 20px or 22px. Length for both target words alternated between 5 characters and 8 characters. The full combination of these factors created 24 conditions, of which 16 had a "correct answer" (i.e., one of the words had a larger font size), and 8 of which did not (i.e., the words were the same font size). This allowed us to observe both instances of factor agreement and disagreement, as well as see which way people leaned at the extreme marginal case where the sizes were equal.

We tested 31 participants, each of whom saw 150 stimuli (6 per each of the 24 conditions described above, as well as 6 engagement tests). While this initially

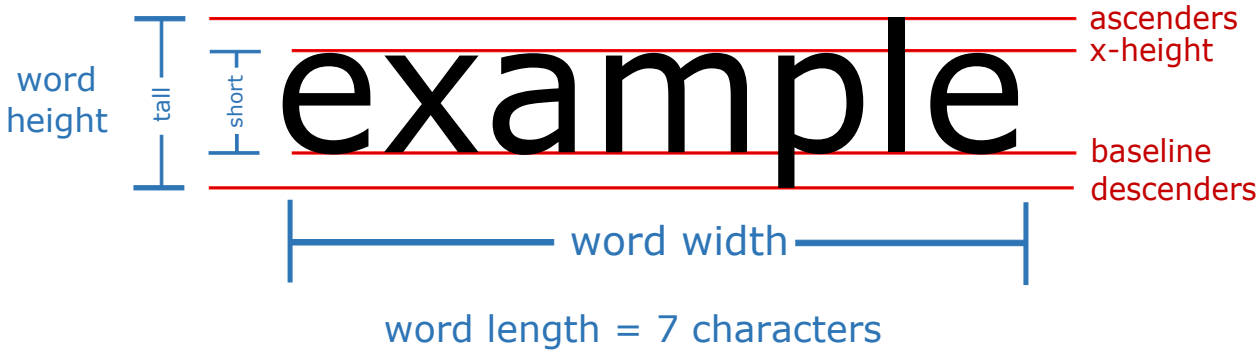



Figure 6.4: We looked for biasing effects on font size perception for three main factors of word shape (shown here in blue): word length (Section 6.4.1), word height (Section 6.4.2), and word width (Section 6.4.3). For our experiments on height, words were broken down into two categories: "tall" words containing both ascenders and descenders, and "short" words whose height was contained between the font's baseline and x-height.

seemed like a large number of stimuli, we saw no fatigue effects in any of our studies. Average time to completion was 5.8 minutes, and the comments we received from participants were positive. We analyzed answers to questions with a correct answer and without a correct answer separately.

For data where there was a correct answer, we calculated the font size difference (1 or 2 px) and word length agreement ("agree," "neutral," or "disagree") for each stimulus. We then ran a two-way analysis of variance (ANOVA) to test for the effect of the font size difference and word length agreement. We saw main effects for both font size difference ($F(1, 150) = 59.21$, $p < 0.0001$) and word length agreement ($F(2, 150) = 14.91$, $p < 0.0001$). Specifically, participant performance decreased when the difference in word length *disagreed* with the difference in font size, as well as when the difference in font size was smaller (see Figure 6.5). A post hoc test using Tukey's HSD showed that the "disagree" condition was significantly different from both the "neutral" and "agree" condition, though the latter two were not statistically distinguishable from one another.

For data where there was no correct answer, we tested to see if the rate at which participants picked the *longer* of the two words was significantly different from chance. Specifically, we calculated the rate at which each participant picked the longer of the two words when the font sizes were the same ($M = 0.59$, $SD = 0.17$) and ran a two-tailed, paired Student's t-test to compare these values against an equally sized collection of values of 50%. We found that participants were significantly more likely to pick the longer of the two words ($t(30) = 2.99$,

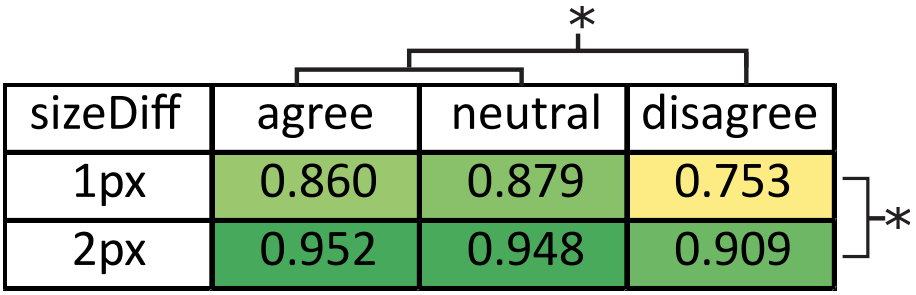



Figure 6.5: This table shows the average participant accuracy for each combination of factors for experiment LEN1 (Section 6.4.1). A two-way ANOVA showed significant main effects for both size difference and length agreement. A post hoc Tukey's HSD test showed that the "disagree" condition (i.e., when the longer of the two words had the smaller font size) was significantly different from the "agree" and "neutral" cases, though the latter two were not distinguishable from one another.

$p = 0.005$), indicating the same direction of bias as seen with the data with correct answers.

**Experiment LEN4**

For this experiment, we wanted to test whether the effects that we had seen using "fake" words and our relatively sparse word clouds would still be present in a more realistic setting. Specifically, rather than generating random strings of characters for words, we used words drawn from the COCA (Davies, 2011). We also switched from our own word cloud implementation (Figure 6.3) to a modified version of a commonly used library called jQCloud (Ongaro, 2014) (Figure 6.6). These clouds packed words more densely by using the spiral positioning layout. The jQCloud library also allowed us to easily modify the aesthetics of the clouds through CSS, creating images more closely resembling word clouds participants might see in other contexts, such as Wordles (Viégas et al., 2009).

Our factors were once again font size and word length, each a within-subject factor by our design. We held the first target word at a font size of 20px while the second word's font size was either 21px, 22px, 23px, or 24px. The word length of each target word alternated between 5 and 8 characters. All words were restricted to characters that contained no ascenders or descenders to avoid any effects resulting from height. The full combination of these factor levels resulted in 16 combinations—each, in this case, with an explicitly correct choice.

We tested 20 participants, each of whom saw 102 stimuli (6 per each of the 16


Figure 6.6: To create a more realistic context for experiment LEN4 (see Section 6.4.1), we used a modified version of the jQCloud library to create stimuli (Ongaro, 2014). These word clouds were more densely packed, more closely resembling what participants might be used to seeing in other settings.

conditions, plus an additional 6 engagement tests). After calculating the font size difference and word length agreement for each stimulus, we ran a two-way ANOVA to test for the effect of these two metrics. Once again, we saw main effects for both font size difference ($F(3, 269) = 7.84$, $p < 0.0001$) and word length agreement ($F(2, 269) = 14.32$, $p < 0.0001$), indicating lower accuracy in instances of word length disagreement at close font sizes (see Figure 6.7). Post hoc tests with Tukey's HSD identify the "disagree" condition and the closest font size difference as the main departures from the rest of the conditions. The lack of difference between the higher-scoring conditions may be the result of ceiling effects, as accuracy was very high across the board.

**Discussion**

In these experiments, we see a very consistent bias towards longer words. Word length, it appears, does affect user perception of font size. However, accuracies across both experiments were higher than we had been anticipating. With mean accuracies consistently near or above 90%, participants seemed surprisingly good at making these comparisons. These high accuracies may have created a ceiling effect, which could account for the lack of distinction between the "agree" and "neutral" conditions in post hoc tests. Dips in accuracy, while consistent, happened

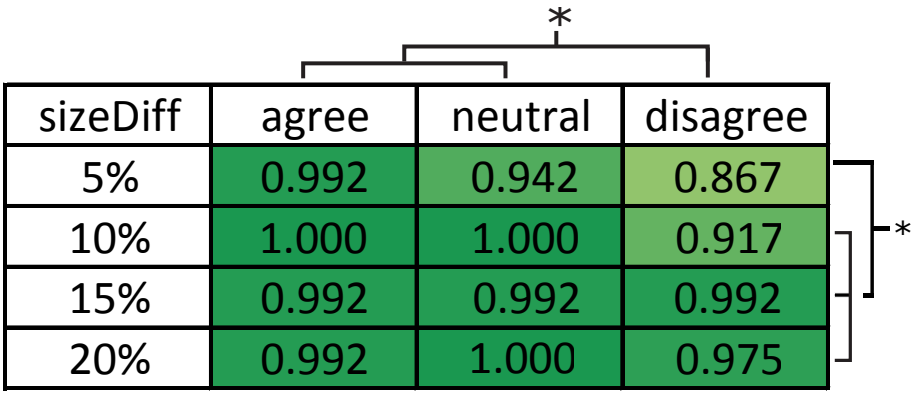



Figure 6.7: This table shows the average participant accuracy for each combination of factors for experiment LEN4 (Section 6.4.1), in which we looked for a bias of length agreement within a more realistic collection of word clouds. After a two-way ANOVA showed significant main effects for both length agreement and font size difference, post hoc tests showed that the "disagree" condition and the closest font size difference were the real departures from the rest of the conditions.

primarily at very close font sizes, but even then participants did notably better than chance. This may be cause to *trust* user perceptions of font size encodings. However, the number of letters is just one of many features that factors into the diversity of shapes words can make.

### 6.4.2 Word Height

The next potentially biasing feature of a word that we tested was a word's *height*. Specifically, there are some characters in the basic Latin alphabet that are taller than others due to the presence of **ascenders** and **descenders** in their glyphs. Ascenders—found for example in the letters "h" and "k"—are marks that reach above a font's *x-height*, while descenders—as in "g" and "y"—extend below a font's baseline (see Figure 6.4). Given that height is perhaps the easiest way to tell font sizes apart when comparing words of varying lengths, we wanted to see whether the presence or lack of such characters would adversely affect user judgment.

We ran five experiments investigating this possibility, and saw a significant bias for character height in each of them (Table 6.1). I will discuss the most important of these experiments here and relegate the others to Appendix A.

#### Experiment HEIGHT1

For our first experiment investigating the effect of character height, we again used words of random characters to give us fine-tuned control over the characters

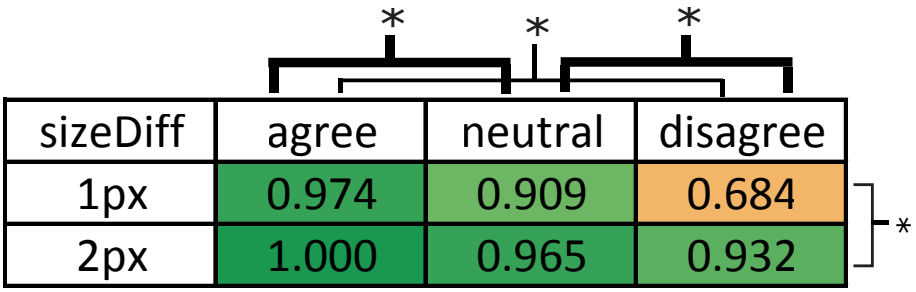



Figure 6.8: This table shows the average participant accuracy for each combination of experimental factors for experiment HEIGHT1 (Section 6.4.2). A two-way ANOVA showed main effects for both word height agreement and font size difference. Post hoc analysis using Tukey's HSD showed that all experimental conditions were statistically distinguishable from one another. Most notably, accuracy is lowest for the "disagree" condition with the closest difference in font size.

present. We defined two types of "fake" words: **tall** and **short**. Short words were generated using only characters without ascenders and descenders (e.g., "a" or "c") and excluding characters of abnormal width (e.g., "w" or "i"). For tall words, we used the vowels "a", "e", "o" and "u" and added characters with ascenders and descenders, again excluding tall characters with abnormal width (e.g., "f", "j", "l"). Short words are naturally rectangular since all of their characters share the same height, but the ascenders and descenders in tall words unbalance this rectangular shape. In order to balance the tall words' shapes, we positioned tall characters both in the beginning and end of the word making sure that if a word started with an ascender, it would end with a descender and vice-versa. Each tall word was made up of 8 characters: 3 short characters and 5 tall characters.

We used the same experimental setup as in Section 6.4.1, with the factor of word length exchanged for word height: the presence or absence of ascending and descending characters. The first target word again varied between sizes of 20px, 21px, and 22px while the second word varied between 20px and 22px as both words alternated back and forth between the tall and short words. Of the 24 conditions created by combining these factors, 16 had a difference of font size (and therefore a "correct" answer) while 8 did not. We analyzed the data for stimuli with a correct answer and stimuli without one separately.

For data where there was a correct answer, we calculated the font size difference (1 or 2 px) and word height agreement ("agree," "neutral," or "disagree") for each stimulus. We then ran a two-way ANOVA to look for effects of these metrics on participant accuracy. We saw significant main effects for both height agreement ($F(2, 155) = 71.22$, $p < 0.0001$) and font size difference ($F(1, 155) = 55.31$,

$p < 0.0001$). These effects went in the same direction as seen in Section 6.4.1 with word length: accuracy dropped when character height *disagreed* with font size and when the font sizes were particularly close (see Figure 6.8). Post hoc tests with Tukey's HSD showed all pairwise combinations of conditions to be statistically significant.

For data without a correct answer, we calculated the rate at which each participant picked the *tall* word when presented with two words of the same font size ($M = 0.67$, $SD = 0.07$) and compared these values to a collection of 50% values with a two-tailed, paired Student's t-test. We saw that participants chose the taller of the two words at a significantly higher rate than chance ($t(31) = 12.91$, $p < 0.0001$).

#### Discussion

Like word length, character height seems to create a consistent bias on participant perception of font size. In fact, the bias for character height seems to be more pronounced, with accuracy in the worst cases dropping to levels not much better than chance (see Table 6.1). However, instances of these height differences are relatively rare in English. The list of words we used from COCA (Davies, 2011) has in total 25,859 eligible words after removing duplicates and words containing numerals and punctuation. Of these, only 870 fit our definition of "short" words—approximately 3.3% of eligible words. As such, the extreme comparison of tall to short words would likely not happen often in the wild. However, there are less extreme comparisons—words containing only a few ascenders or descenders, words containing only one or the other, etc.—that may be more common and still exhibit this bias.

### 6.4.3 Word Width

After running our tests on word height, we decided to look for the the effect of a different factor: word *width*. In our height experiments, we held length constant and attempted to control for width by excluding characters of abnormally small or large width (as described in Section 6.4.2). However, there were still small differences in glyph widths even outside of those characters, which created variance in width from word to word, even within the same length conditions. In a post hoc test, we computed a **width agreement** metric for each stimulus from experiment HEIGHT2 indicating whether the difference in width went in the same direction as the difference in font size. It was only for stimuli with the

smallest font size difference that we saw any width disagreement, given that we had attempted to make widths neutral. We ran a two-way ANOVA looking for an effect of width agreement, specifically on the stimuli in the closest font difference case. The effect we saw was significant ($F(2, 38) = 13.73$,$p < 0.0001$). Accuracy in the disagree condition ($M = 0.523$, $SD = 0.18$) was substantially lower than accuracy in the agree condition ($M = 0.82$, $SD = 0.10$).

This begs an interesting question. We knew that longer words created a bias for font size perception, as described in Section 6.4.1, but we did not know *why*. Was this bias the result of longer words taking up more space, and therefore a function of width, or were participants actually making a numerosity judgment about the letters? We hypothesized that the main factor in this effect was width rather than length, thinking that words—especially *real* ones—are generally read as a whole, rather than letter by letter (Haber et al., 1983). To test this hypothesis, we ran two additional experiments isolating the effects of width and length.

**Experiment WIDTH1**

In our first of these experiments, we wanted to see whether word width biased font size perception even when the number of characters and character height were held constant. Varying width but not length put a tight constraint upon the words we were able to use; differences between character widths are small, and so words that differ substantially in one factor but not the other are rare. For our stimuli, we chose a collection of pairs of words that were each 8 characters long, but differed in **raw width** by 10 pixels. We defined "raw width" to be a word's width computed at a font size of 20px, so that we could have a measure of width differences that was separate from our font size factor. We also made sure that each pair of words shared the same character height.

Our two factors for this experiment were width agreement and font size difference. For each stimulus, one of the target words had a font size of 20px, while the other was either 21px, 22px, 23px, or 24px. For the width agreement factor, the larger of the two words either had a raw width that was 10 pixels greater than the smaller word ("agree") or 10 pixels less than the smaller word ("disagree"). Four font size differences combined with two levels of width agreement gave us 8 conditions, each of which had a "correct" answer.

We tested 20 participants, each of whom saw 56 stimuli (6 per each of the 8 conditions, as well as 6 engagement tests). After calculating the font size difference and width agreement of each stimulus, we ran a two-way ANOVA to

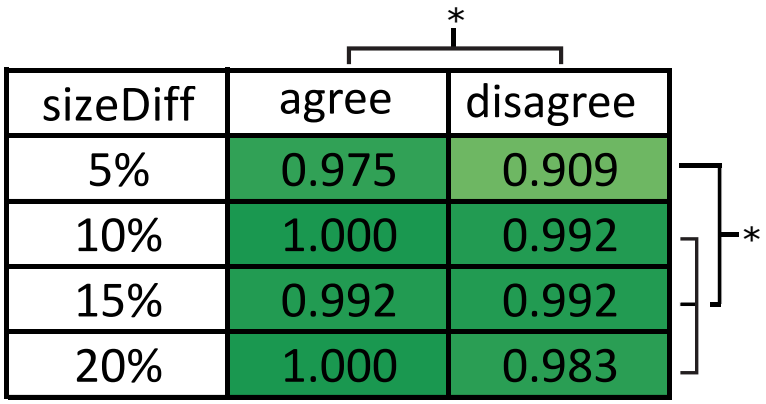



Figure 6.9: This table shows the average participant accuracy for each combination of experimental factors for experiment WIDTH1 (Section 6.4.3). In this experiment, target words had a difference of 10 pixels in raw width (i.e., their width at the same font size). In the "agree" condition, this width difference was in the same direction as the difference in font size, while it was in the opposite direction for the "disagree" condition. A two-way ANOVA showed significant main effects for both width agreement and font size difference. Only the lowest size difference was statistically distinguishable in post hoc tests, perhaps due to ceiling effects given the very high overall accuracy.

test for the effects of the two factors on participant accuracy. We saw main effects for both width agreement ($F(1, 133) = 11.33$, $p = 0.001$) and font size difference ($F(3, 133) = 6.77$, $p = 0.0003$) indicating a drop off in accuracy for width disagreement at close font sizes (see Figure 6.9). While a post hoc Tukey's HSD test only showed the smallest size difference condition to be statistically distinguishable, this may have been due to ceiling effects, given the very high accuracy across all other conditions.

**Experiment WIDTH2**

In the second of these experiments, we wanted to see whether the number of letters in a word had any effect on font size perception outside of the correlated factor of width difference. For our stimuli, we chose pairs of words that had the same raw width (described in Section 6.4.3) but differed by 3 letters in length. Of the words we had available from which to choose, this was the largest length difference that provided us with enough pairs. Each pair of words shared the same character height, as well.

Our two factors for this experiment were length agreement and font size difference. Once again, one of the two target words in each stimulus had a font size of 20px, while the other was either 21px, 22px, 23px, or 24px. For the length

| sizeDiff | agree | disagree |
|---|---|---|
| 5% | 0.982 | 0.982 |
| 10% | 1.000 | 0.991 |
| 15% | 0.991 | 1.000 |
| 20% | 0.982 | 1.000 |

Figure 6.10: This table shows the average participant accuracy for each combination of experimental factors for experiment WIDTH2 (Section 6.4.3). In this experiment, target words had a difference of 3 characters in their length (going with or against the direction of the difference in font size in the "agree" and "disagree" conditions, respectively). A two-way ANOVA showed no significant main effects for either factor, and accuracy was very high across the board.

agreement factor, the larger of the two words had either 3 more characters than the smaller word ("agree") or 3 fewer characters than the smaller word ("disagree"). Four font size differences combined with two levels of length agreement gave us 8 conditions, each of which had a "correct" answer.

We tested 19 participants, each of whom again saw 56 stimuli. After computing the font size difference and length agreement of each stimulus, we ran a two-way ANOVA to test for the effects of these factors on participant accuracy. This time, we saw no main effects for either font size difference ($F(3, 126) = 1.47$, $p = 0.23$) or length agreement ($F(1, 126) = 0.00$, $p = 1.00$). Accuracy was quite high across all conditions (see Figure 6.10). This seems to indicate that any bias created by number of letters alone is not strong enough to register without also varying the stronger factor of word width.

**Discussion**

The restriction of varying only width *or* length meant that we were not able to test very large differences in either factor. As such, we did not expect to see a vary large effect size for either experiment. However, from these results, we feel we can conclude that width is the more important factor to consider when worrying about bias. Length may matter in some extreme cases, but we stretched the degree to which length can vary without width to the limits of the English language, and still saw no effect. Practically, therefore, width seems the more relevant concern.

## 6.5 Debiasing with rectangles

In Section 6.4, we show that there are multiple ways in which a word's shape can bias interpretation of its font size. Depending on the task a designer intends a user to undertake, the effect of this bias may not be large enough to warrant much intervention—a possibility we discuss further in Section 6.7. However, for tasks precise enough to be concerned by these effects, the next question is what we can do as designers to *mitigate* this bias.

One potential method for this debiasing effort was inspired by the work of Correll et al. debiasing area discrepancies in tagged text (Correll et al., 2013). In this work, the authors determined that users suffered from an area bias when making numerosity judgments of words tagged with colored backgrounds. Specifically, when the number of words disagreed with the *area* of the colored backgrounds, accuracy dropped dramatically. However, they were able to counteract this bias by adjusting the area of the backgrounds for underrepresented words.

We suspected that such a technique could be useful for the biases we observed in font size encodings. By enclosing individual words in filled bounding boxes, we can create a redundant encoding for font size that may alleviate the issue of diverse word shapes. These bounding boxes would also give us a glyph whose proportions we can adjust without fearing any change in legibility.

As such, we decided upon the following potential debiasing technique: We would surround each word with a **padded bounding box**. These boxes would contain the full height of any potential character, going from the ascender line to the descender line (see Figure 6.4). The width of each box would be adjusted such that they all shared the same raw width—which is to say, they would be equal in width if they all contained words of the same font size. With such padding, the difference in rectangle width and height would always agree with the font size difference for any two words, creating a more reliable and readable indication than the word alone. We ran an experiment to test whether this strategy would help increase user accuracy in cases of factor disagreement.

### 6.5.1 Experiment BOX1

To test our debiasing technique, we ran an experiment with a similar design to that described in experiment LEN4 (described in Section 6.4.1). The factors for our stimuli were font size difference (which varied in increments of 5, 10, 15, and 20% from a base font of 20px) and word length (which alternated between 5 and 8


Figure 6.11: By containing each word in a color-filled bounding box and padding the sides of each bounding box such that their widths were proportional to their font sizes, we were able to eliminate the effect of width disagreement.

characters for each word). For this experiment, we also ensured that whenever the two target words were the same length, they also had the same raw width, and when they were not the same length, they had a difference in raw width of 20 pixels. These factor levels created 16 conditions, each with a "correct" answer.

Rather than showing participants a pure word cloud, we placed padded bounding boxes around each word (see Figure 6.11). These bounding boxes were padded on either side such that the rectangle for each word had the same raw width before any differences in font size had been applied. Participants were instructed in the tutorial that the rectangles containing the words were sized proportionally to the words' font sizes.

We tested 20 participants, each of whom saw 102 stimuli (6 for each of the 16 conditions, plus an additional 6 engagement checks). After computing the length/width agreement and font size difference of each stimuli, we ran a two-way ANOVA to test for the effects of these factors on participant accuracy. While we found a significant main effect for font size difference as before ($F(3, 209) = 10.88$, $p < 0.001$), we saw no effect of length/width agreement ($F(2, 209) = 0.52$, $p = 0.60$). Even in the typical worst case—conditions with factor disagreement and the smallest difference in font size—participants scored over 90% accuracy (see Figure 6.12). To this degree, it seems that the padded bounding boxes were successful at mitigating the bias introduced by length/width disagreement.

| sizeDiff | agree | neutral | disagree |
|---|---|---|---|
| 5% | 0.914 | 0.932 | 0.908 |
| 10% | 0.983 | 0.992 | 0.933 |
| 15% | 0.983 | 0.971 | 0.992 |
| 20% | 0.992 | 0.996 | 0.983 |

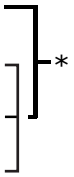



Figure 6.12: This table shows the average participant accuracy for each combination of experimental factors for experiment BOX1 (Section 6.5.1). In this experiment, words were given padded bounding boxes (as in Figure 6.11) in an attempt to mitigate the bias created by disagreement in word width. While a two-way ANOVA showed there to be a significant main effect of size difference on accuracy, no main effect was seen on word width agreement—indicating that padded bounding boxes may be a viable way of debiasing font size perception.

This technique of debiasing font size encodings is primarily a proof-of-concept. Aesthetically, word clouds like the one in Figure 6.11 are inferior to more standard layouts, and aesthetics can be an important factor to an encoding's utility (van der Geest and van Dongelen, 2009). It may be possible to create more aesthetic approaches, perhaps using other word features like font weight or tracking. At any rate, this shows that the effects of word shape on font size perception are possible to correct for.

## 6.6 Alternate task

A possible critique of this work is that our experimental task (pick the bigger of two highlighted words) does not necessarily reflect how font size encodings are used in the wild. Our reason for using this task was that it acts as a "visual primitive" for broader, more general tasks (see Section 6.2). It is not our intention to say that people routinely have to perform the act of comparing two words within a word cloud, but rather that the more high-level, interpretation-based tasks that people *do* perform (such as gist-forming as described in Chapter 5) rely upon this low-level perceptual ability.

Nonetheless, we wanted to confirm that the bias that we saw within the compare-two-words task was not specific to this precise experimental setup. In a further set of experiments, we looked for the same bias using a different task: finding the single biggest word within a cloud. While we believe that this task relies upon the same perceptual abilities as the comparison task, it is in some ways closer to how

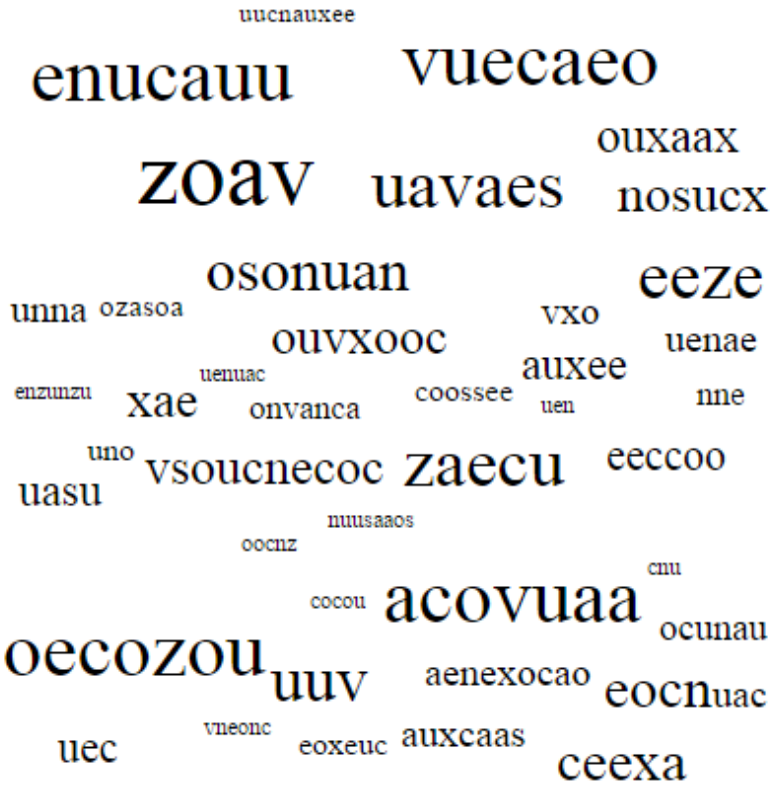



Figure 6.13: For experiments BIG1 (Section 6.6.1) and BIG2 (Section 6.6.2), participants were presented with word clouds of pseudowords and asked to pick the one with the biggest font size. In this example, "zoav" is the correct answer, with four near misses that are of longer length.

word clouds are used in practice. Picking out the biggest word (or words) from a font size visualization is similar to the higher level task of asking what the data encoded by the visualization is "about."

To give us control over the gap in font size between target words similar to what we had in our previous experiments, we introduced a concept called **near misses**. Near misses are words that are *almost* as large as the biggest font size word, but not quite (see Figure 6.13). Explicitly controlling the near misses in each stimulus allowed us to evaluate multiple font size differences between the biggest word and the next biggest. It also gave us a new factor: the number of near misses.

Our general hypotheses for the pick-the-biggest task were that participant accuracy would be worse in instances of factor disagreement (as in our previous experiments), and that this effect would be more pronounced in stimuli that contained *more* near misses to distract the participant.

### 6.6.1 Experiment BIG1

In our first experiment making use of the pick-the-biggest task, we sought to examine potential bias due to word length agreement or disagreement. We created a set of stimuli of word clouds made up of pseudowords (see Section 6.3.3). As before, stimuli contained 40 distractor words, in this case limited to font sizes below 40px. Stimuli then contained either 1 or 4 near miss words which were given a font size of 40px. Finally, each stimulus contained a target word (the "correct" choice) with a font size defined by a percentage increment above that of the near misses (either 5, 10, 15, or 20% bigger).

The factors for this experiment were font size difference (5, 10, 15, or 20%), target word length (5 or 8 letters), near miss word length (5 or 8 letters), and number of near misses (1 or 4). Each factor was varied within participants. The full combination of these factor levels resulted in 32 conditions. We tested 19 participants, each of whom saw 134 stimuli (4 per each of the 32 conditions, plus an additional 6 engagement tests with a font size difference of 50%). After calculating font size difference and word length agreement for each stimulus, we ran a two-way ANOVA to test for the effect of the three metrics (including number of near misses). We saw main effects for all three factors: font size difference ($F(3, 414) = 5.82$, $p = 0.0007$), length agreement ($F(2, 414) = 10.10$, $p < 0.0001$), and number of near misses ($F(1, 414) = 33.66$, $p < 0.0001$), indicating lower accuracy in instances of word length disagreement, more near misses, and closer font sizes (see Figure 6.14).

Our hypothesis that we would still see a biasing effect of length disagreement using a different task was confirmed. Interestingly, accuracies seemed to drop off even more when participants were performing the pick-the-biggest task than when they were performing the pairwise comparison task (see Figure 6.14). However, participants still achieved greater than 50% accuracy in each condition, performing better than chance.

### 6.6.2 Experiment BIG2

For a second experiment using the pick-the-biggest task, we were interested in whether the *magnitude* of the word length agreement or disagreement was relevant to the bias created—that is, would instances of greater disagreement hurt accuracy more than instances of small disagreement. We created a design that was similar to that described in Section 6.6.1, but with different levels for the word length

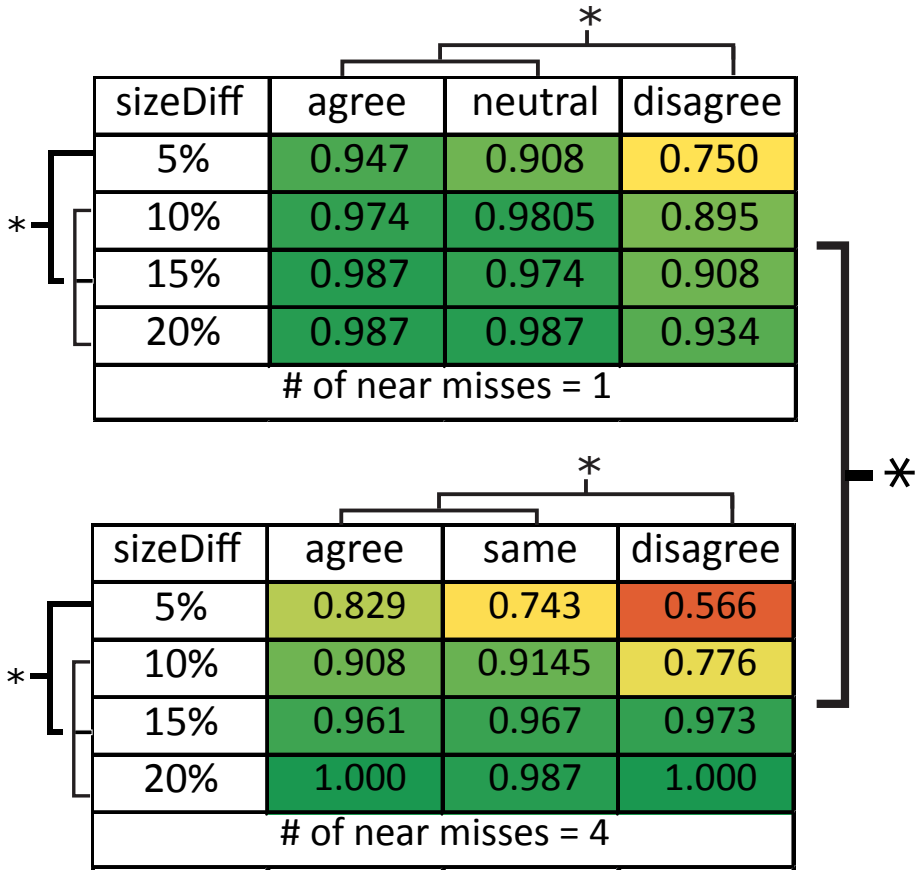



Figure 6.14: This table shows the average participant accuracy for each combination of experimental factors for experiment BIG1 (Section 6.6.1). In this experiment, participants were asked to select the word with the largest font size. They were presented with word clouds containing a single word bigger than the rest (the "target" word) along with either 1 or 4 "near misses." A two-way ANOVA showed there to be a significant main effect for both the font size difference between the target and the near misses, for word length agreement, and for the number of near misses.

disagreement factor. Rather than only considering words of 5 or 8 characters, we considered word length differences of 1, 3, and 5 characters in both the "agree" and "disagree" directions, for a total of 6 levels for this factor. We hypothesized that instances of large disagreement (e.g., 5 characters) would show lower accuracy than instances of small disagreement (e.g., 1 character).

We tested 19 participants, each of whom saw 150 stimuli (3 per each of the 48 combinations of factors with an additional 6 engagement checks). We ran a two-way ANOVA to test for the effects of the three metrics, and again saw main effects for all three: font size difference ($F(3, 846) = 3.02$, $p = 0.03$), length difference ($F(5, 846) = 8.00$, $p < 0.0001$), and number of near misses ($F(1, 846) = 7.00$,

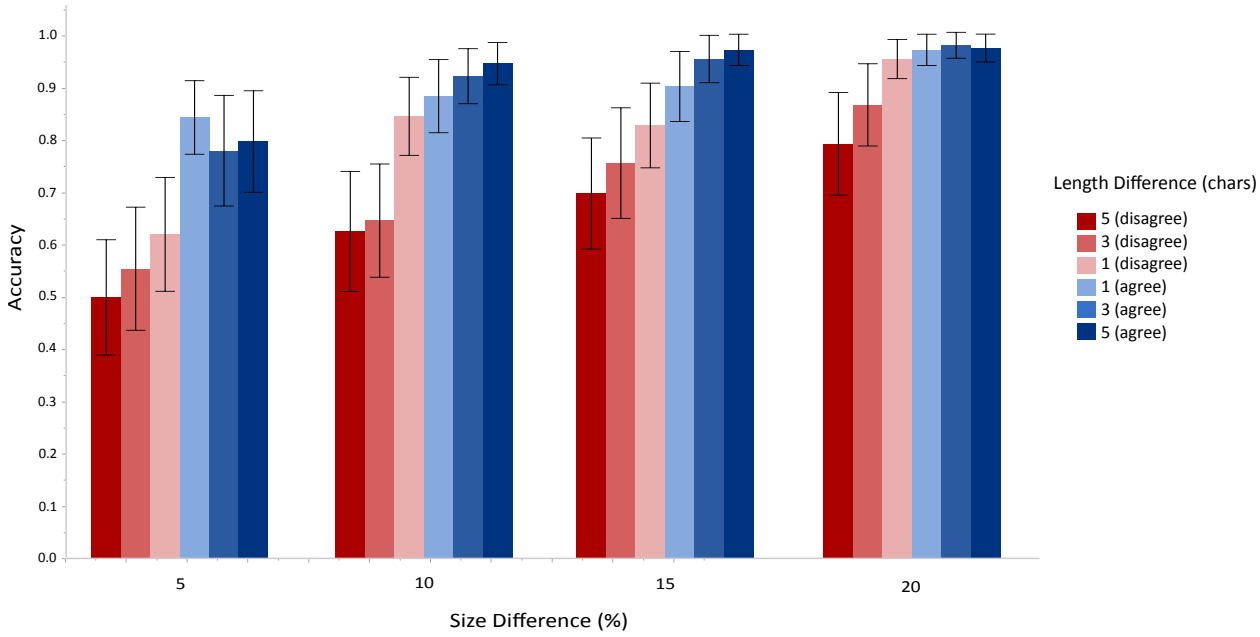



Figure 6.15: This graph shows the average participant accuracy for combinations of experimental factors in experiment BIG2 (Section 6.6.2). In this experiment, participants were tasked with picking the word with the largest font size as in Section 6.6.1. We tested a wider variety of length differences, and saw that performance was generally lowest in cases of large disagreement and highest in cases of large agreement. These values are averaged across two levels of the "number of near misses" factor. Error bars represent a 95% confidence interval.

$p < 0.008$)—each in the same direction as seen previously. We also noted, as expected, that accuracies were lowest in instances of largest disagreement and highest in instances of largest agreement (see Figure 6.15).

### 6.6.3 Discussion

The main takeaway from these two additional experiments is that the biasing effect of factor disagreement is not isolated specifically to the task of pairwise comparison, but can also be seen in a task that specifically tries to draw the user's attention to the most "important" word in the visualization. The detrimental effect of more "near misses" seems to perhaps indicate that while people are generally able to perform pairwise comparisons, needing to perform *multiple* of these can cause them to miss smaller words. However, performance is still considerably better than chance in all but the most pathological cases.

## 6.7 Full discussion

Results from experiments not described above are laid out in Appendix A. In those experiments, we looked for a number of extra details and effects. We compared performance at different base font sizes. We tested to see if the results were the same with a sans serif font (which they were). We looked for a size difference ceiling past which participant accuracy maxed out (which was between 20-25% size difference). Consistent across each experiment were the same things we saw in each of the experiments described in Sections 6.4, 6.5, and 6.6: decreased performance with factor disagreement at close size differences. It is worth noting that this effect is not simply the result of participants focusing on *area* rather than font size. Consider examples from our length disagreement experiments. While we observed decreased accuracy when a word with a 1-pixel-larger font size was significantly shorter than the other target, increasing the font size difference a mere pixel more resulted in very high accuracy—even though the difference in area disagreement created by this change in font size would be minimal.

Clearly, perceptions of font size can be biased by these factors. The relevant question for a designer is how much this bias will affect their end users, and whether it is worth designing around it. The largest effects that we saw occurred at very close differences in font size, and even then participants performed better than chance. Despite the fact that font size encodings are rarely used for tasks requiring pixel-level accuracy, our findings seem to suggest that they may be more suitable for such tasks than previously thought. Given the particular utility of the font size encoding for textual data, expanding its potential uses could have significant impact. An important future direction of this work is to continue testing the limits of this perception in real-world applications.

Our debiasing attempts are a proof-of-concept, and show that it is possible to correct for the effects of factor disagreement if a designer expects careful reading and comparison of their encodings. We believe there are more aesthetic ways of making these corrections, which should be explored further. Font weight, for instance, may interact with font size in ways that could be exploited. Possible candidates for other methods include typeface modifications such as kerning, widths of individual letter glyphs, or even exploring the use of monospaced typeface (where all the characters have the same width causing words that have the same length to be the same width as well). Ultimately, whether or not debiasing is even necessary depends on how the encoding will be used in practice.

While we tested a number of features related to a word's content for their biasing effects—including length, width, character height, and to a lesser degree font (see Appendix A)—there are more features that could be examined. These include color, font weight, and a word's semantic weight or meaning. Also, while we believe that the pairwise comparison and pick-the-biggest tasks allow us to get down to the perceptual primitives of higher level tasks, there are many other tasks to be considered in order to better understand font size encodings in real world contexts.

In this chapter, I have explored the effects of different word shapes on the perceptual ability to judge data encoded through font size. While the consistent, statistically significant biases of factor agreement are the most obvious takeaway, the one that may have the greatest impact is the consistently high performance of participants at these high-precision tasks. We entered into this work with the hypothesis that font size encodings were only useful for tasks that did not require much precision, and much of the field would dispute that they are even good at that. However, participant performance was consistently impressive across a wide variety of intentionally difficult factors and conditions. Though there is much more work to be done to better understand the constraints upon this performance, this may eventually open up font size encodings for a much wider range of tasks.

Regarding conveying topics within topic model visualizations, these results offer further validation for the use of word clouds for topic representation, indicating that users are in fact able to accurately perceive font-size-encoded values so as to factor them into their understanding of the distribution. This work is illustrative of reductionist evaluation that can be used to both validate the specific use of an encoding while also building our understanding of that encoding in other contexts. Ultimately, this generalizability has the potential to lead to greater impact.

# 7 CONCLUSION

*Now my charms are all o'erthrown,*
*And what strength I have 's mine own,*
*Which is most faint. Now 'tis true*
*I must be here confined by you,*
*Or sent to Naples. Let me not,*
*Since I have my dukedom got*
*And pardoned the deceiver, dwell*
*In this bare island by your spell,*
*But release me from my bands*
*With the help of your good hands.*
*Gentle breath of yours my sails*
*Must fill, or else my project fails,*
*Which was to please. Now I want*
*Spirits to enforce, art to enchant,*
*And my ending is despair,*
*Unless I be relieved by prayer,*
*Which pierces so that it assaults*
*Mercy itself, and frees all faults.*
*As you from crimes would pardoned be,*
*Let your indulgence set me free.*

— Prospero, *The Tempest*

Topic modeling is a powerful tool for large-scale text analysis. It can help researchers uncover new insights in otherwise overwhelming amounts of sheer text. However, understanding the complexity of a topic model can be difficult. Visual tools and techniques can help scholars from a wide variety of domains unpack this complexity. In this dissertation, I have presented a method for using topic models that enables visual exploration of them in a way that ties high level patterns to low level exemplars, merging the processes of distant and close reading and giving researchers the ability to make sense of the model using skills and language they understand.

To illustrate and validate this method, I first presented a system designed to afford this sort of inquiry and then supported the system in a variety of ways. This support involved taking a step back from a single model to consider comparison *between* models as a necessary ability for researchers deciding which models to investigate. I have also supported the encodings used within my approach with empirical analysis and validation, particularly those of topic encodings and font size. It is my hope that I have both provided a system and techniques that can be used for topic modeling and text exploration in a variety of domains, as well as generalizable knowledge about topic representations and font size encodings that can inform other design.

## 7.1 Limitations

There are a variety of limitations and unanswered questions associated with the work presented here. Many such are described within their corresponding chapters. Here, I will lay out what I believe to be the most significant.

**Scaling up** While the reorderable matrix described in Chapter 3 offers a high level view of large corpora, this view is nonetheless limited. To see the most global trends, many researchers want to be able to see a "map" of the documents, fitting them all into a single picture in a way that reflects their relationships with each other. While we discuss such a view in Section 3.3.2, it is not as well integrated into the workflow as the other views, and could be improved to provide more interaction into the clusterings and similarities it may uncover. A number of methods and tools exist for placing documents into two-dimensional embeddings (Correll et al., 2011; Endert et al., 2012a; Stasko et al., 2008). Better incorporating

such a view on top of those existing in Serendip could give researchers even more freedom to investigate hypotheses across levels of abstraction.

**Additional comparison tasks** Though we believe that the comparison tasks we outlined in Chapter 4 provide good coverage over the typical uses of topic modeling, there are still a number of tasks that are unsupported by the techniques we laid out. Most notably, though we provide comparison at the document level, I believe that it will ultimately be important to provide comparison at the passage level, as well. Tagged text encodings to highlight key differences within passages could help researchers appreciate the contrasts between two models in the context of the lines with which they are most familiar.

**Topic model specificity** The tools and techniques presented in this dissertation are focused specifically on topic models, limiting the scope of their impact. However, they need not be. I believe that these techniques can and should be extended for use with arbitrary model types. This has already been done for *pieces* of the work presented here. For example, TextViewer (Chapter 3) has been incorporated into a tool called Ubiq that lets users tag their own documents using either dictionaries of linguistic categories (Collins and Kaufer, 2001) or their own schemas. Users can explore arbitrarily tagged documents to find and understand inner-document trends. The rest of the levels described in Chapter 3 can also be made to enable exploration of any vector-based models of text. Similarly, buddy plots (Chapter 4) have the potential to serve well as comparison tools for arbitrary sets of distance functions.

**Model training** Perhaps the biggest limitation of this work is the lack of support for interactive model training. While we provide the techniques to allow researchers to incorporate their interest and domain expertise into the exploration of the topic model, it is still difficult for them to use that expertise to *create* the perfect model for answering their questions. Model comparison helps address this deficiency, but pairwise comparison does not yet scale to the full parameter space. More work needs to be done to involve researchers in the full process of curation, training, and discovery.

**Assessment of split topics** While Chapter 5 showed the robustness of people's ability to form gists of topics as presented through multiple representations, it did

not address their performance with topics that may contain more than a single concept or idea. Such topics can occur especially when the number of topics parameter is set too low to capture the full complexity of a set of documents. It will be important to investigate whether the combination of multiple gists has an effect on user interpretations.

## 7.2 Future work

While this dissertation answers many questions, it opens up even more. There are many future projects that could build on the work I have presented here.

**Interactive model building** While the challenge of training models is a limitation of this work, it is also one of its most interesting open questions. It may be that scaling up model comparisons to deal with more points in the parameter space would serve users well. However, it is possible that rather than asking users to build many models and compare them, we would be able to get their input incrementally during the training of a single model, effectively mining them for their expertise in such a way that the final model reflects their background knowledge. As work progresses on efficient methods for building models on incremental data (Yao et al., 2009), it may be possible to incorporate user input, perhaps by directing the model to the most relevant documents or by providing semi-supervised tuning along the way. Other work has looked at incorporating expert knowledge in the form of first-order logic (Andrzejewski et al., 2011). While expressing knowledge in this form may be difficult for some domain researchers, this could be overcome with the proper visual interactions.

**Document structure** While I believe I have laid out good techniques for finding similarities within document proportions, there is still much to be explored to help researchers compare documents by their structure. It would be useful to help researchers answer questions such as "What documents exhibit these topics at the beginning and these other topics at the end?" There has been some work on using algorithms from signal processing to compare word trends (Correll and Gleicher, 2016). This could be valuable in the context of topic model exploration.

**Optimizing document chunking** Related to better methods of comparing document structures are better methods of document *chunking*. As described in

Chapter 4, chunking is a common pre-processing step that can build finer grain models which give insight into the rise and fall of topics within documents. However, precisely *where* to cut documents apart is an open question. While some documents have pre-defined semantic breaks (e.g., scenes or chapters), many do not. The most common practice for chunking is to separate documents at a constant interval length (e.g., 1000 tokens). However, this has the potential to break documents at places of high topic density, dampening or even hiding the importance of particular passages. It may be possible to incorporate researchers into the chunking process in an interactive way, building off the future work above. Other possible methods include overlapping chunks and algorithmic analysis looking for scene breaks.

**Constraining the use of font size encodings** The results presented in Chapter 6 are promising for the use of font size encodings, but they leave many more unanswered questions. One such is how semantics factor into the processing of such encodings. If a word's meaning is unrelated to those surrounding it, is it less likely to be seen? If so, is this a problem with the encoding or simply confirmation bias? There are also questions to be answered about the effect of juxtaposition. If a word is placed next to a smaller one, will its font size be interpreted as bigger?

It will also be important to stretch the limits of font size encodings for other tasks. If they *can* be used in higher-precision tasks, it may require building better, more aesthetic ways of correcting the (small) biases that exist. To determine this, it would also be important to measure the ways in which these biases affect higher level tasks, if the do at all.

The work presented in this dissertation illustrates the viability of visual topic model exploration that connects high level hypothesis formation with explanatory dives into the underlying data. Through my intensive collaboration with domain scholars, I have seen the value of this approach first hand, and I have laid out evidence showing its effectiveness in their research practices. I have additionally validated encodings used by my approach in a way that can inform other visualization design. While there is much further work to be done to fully realize the promise of end-to-end visual curation, training, and discovery within models, my hope is that this work pushes us closer to that goal.

# A ADDITIONAL FONT SIZE EXPERIMENT DATA

*O, do not wish one more!*

— King Henry, *Henry V*

In this appendix, I am including data for the full set of experiments described in Chapter 6 (and outlined in Tables 6.1 and 6.2). This will include brief explanations of the conditions for each experiment and their results, along with breakdowns of the data for each combination of factors and analysis of variance (ANOVA) tables.

# A.1 Length agreement experiments

## A.1.1 Experiment LEN1

The main factors of LEN1 were word length and font size. We tested to see if the number of characters in a word would bias perception. Table A.1 shows the accuracy of word length difference and font size difference. From this experiment, we observed that accuracy drops when font size difference is small and when font size and word length disagree.

We did a two-way analysis of variance (ANOVA) to look for effects of these factors on participant accuracy. In table A.2, we saw that word length creates a significant bias to the font size. Moreover, the table shows that font size difference of two words is a significant factor.

| sizeDiff | agree | neutral | disagree |
|---|---|---|---|
| 1 | 0.86 | 0.88 | 0.75 |
| 2 | 0.95 | 0.95 | 0.91 |

Table A.1: Breakdown Table for LEN1

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiff | 1 | 1 | 150 | 59.2063 | <.0001 |
| lenAgree | 2 | 2 | 150 | 14.9124 | <.0001 |
| sizeDiff*lenAgree | 2 | 2 | 150 | 3.0716 | 0.0493 |

Table A.2: ANOVA Table for LEN1

## A.1.2 Experiment LEN2

In LEN2, the main factors were word length and font size. We tested to see whether larger word length differences would have stronger effects. We also tested the length effect with different font size baselines between-subjects (i.e., was the smaller of the two words 20px or 30px). In table A.3, we can see the accuracy

drops when word length disagrees with font size and when font size difference is small.

We did a two-way ANOVA statistical test to measure the results of LEN2. In table A.4, all three factors (size difference, length agreement, and baseline) are significant. It is worth noting that since baseline was a between-subjects factor, there may be some interference from differences between browsers.

| baseline | sizeDiff% | agree | neutral | disagree |
|---|---|---|---|---|
| 20 | 5% | 0.95 | 0.93 | 0.85 |
| 20 | 10% | 0.98 | 0.98 | 0.95 |
| 20 | 15% | 0.99 | 0.98 | 0.97 |
| 20 | 20% | 1 | 1 | 0.96 |
| 30 | 5% | 0.77 | 0.7 | 0.61 |
| 30 | 10% | 0.94 | 0.95 | 0.87 |
| 30 | 15% | 0.99 | 0.98 | 0.94 |
| 30 | 20% | 1 | 1 | 0.92 |

Table A.3: Breakdown Table for LEN2

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiff% | 3 | 3 | 418 | 58.9634 | <.0001 |
| lenAgree | 2 | 2 | 418 | 12.1342 | <.0001 |
| sizeDiff%*lenAgree | 6 | 6 | 418 | 1.2480 | 0.2806 |
| baseline | 1 | 1 | 37 | 7.9799 | 0.0076 |

Table A.4: ANOVA Table for LEN2

### A.1.3 Experiment LEN3

The main factors of LEN3 were word length and font size. In the previous tests, we observed that 30px had worse accuracy than 20px. Thus, we tested with multiple baselines to see whether accuracy drops when the baseline is larger. Unlike LEN2, baseline here was a within-subjects factor. In table A.5, the results show that we cannot see a clear linear relationship between different baselines.

We did a two-way ANOVA statistical test to measure the result of LEN3 as well. In table A.6, all three factors (size difference, length agreement, and baseline) are significant.

| sizeDiff% | baseline | agree | neutral | disagree |
|---|---|---|---|---|
| 5% | 20 | 0.87 | 0.93 | 0.82 |
| 10% | 20 | 0.98 | 0.93 | 0.98 |
| 15% | 20 | 0.97 | 1 | 0.98 |
| 20% | 20 | 1 | 1 | 1 |
| 5% | 25 | 0.68 | 0.65 | 0.48 |
| 10% | 25 | 0.95 | 0.98 | 0.93 |
| 15% | 25 | 1 | 1 | 0.95 |
| 20% | 25 | 1 | 1 | 1 |
| 5% | 30 | 0.8 | 0.83 | 0.62 |
| 10% | 30 | 0.9 | 0.95 | 0.77 |
| 15% | 30 | 1 | 0.98 | 0.95 |
| 20% | 30 | 1 | 1 | 1 |
| 5% | 35 | 0.95 | 0.95 | 0.65 |
| 10% | 35 | 1 | 1 | 0.9 |
| 15% | 35 | 1 | 0.98 | 0.88 |
| 20% | 35 | 1 | 1 | 0.98 |

Table A.5: Breakdown Table for LEN3

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| baseline | 3 | 3 | 926 | 8.5716 | <.0001 |
| sizeDiffPercent | 3 | 3 | 926 | 85.4334 | <.0001 |
| lenAgree | 2 | 2 | 926 | 31.5987 | <.0001 |
| sizeDiffPercent*lenAgree | 6 | 6 | 926 | 5.5771 | <.0001 |

Table A.6: ANOVA Table for LEN3

### A.1.4 Experiment LEN4

The main factors in LEN4 were word length and font size. In LEN4, we used word clouds filled with English words to see if they exhibited the same effect as the pseudowords used in previous experiments. In table A.7, we observe an effect in the same direction as before: accuracy decreases when length and font size disagree in the real word.

The two-way ANOVA, shown in table A.8, shows that font size difference and length agreement are both significant factors.

| sizeDiff% | agree | neutral | disagree |
|---|---|---|---|
| 5% | 0.99 | 0.94 | 0.87 |
| 10% | 1 | 1 | 0.92 |
| 15% | 0.99 | 0.99 | 0.99 |
| 20% | 0.99 | 1 | 0.98 |

Table A.7: Breakdown Table for LEN4

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiff% | 3 | 3 | 269 | 7.8359 | <.0001 |
| lenAgree | 2 | 2 | 269 | 14.3156 | <.0001 |
| sizeDiff%*lenAgree | 6 | 6 | 269 | 3.8008 | 0.0012 |

Table A.8: ANOVA Table for LEN4

## A.2 Height agreement experiments

### A.2.1 Experiment HEIGHT1

The main factors of HEIGHT1 were character height and font size. In HEIGHT1, we tested to see whether the appearance of ascenders and descenders would result in perceptual bias. In table A.9, accuracy drops when character height disagrees with font size, especially when size difference is small. It shows that tall words (i.e., words containing ascenders and descenders) appear larger than short words. We can further see in table A.10 that both font size difference and character height agreement are significant factors.

| sizeDiff | agree | neutral | disagree |
|---|---|---|---|
| 1 | 0.97 | 0.91 | 0.68 |
| 2 | 1 | 0.97 | 0.93 |

Table A.9: Breakdown Table for HEIGHT1

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiff | 1 | 1 | 155 | 55.3093 | <.0001 |
| heightAgree | 2 | 2 | 155 | 71.2212 | <.0001 |
| sizeDiff*adAgree | 2 | 2 | 155 | 22.7248 | <.0001 |

Table A.10: ANOVA Table for HEIGHT1

### A.2.2 Experiment HEIGHT2

The main factors of HEIGHT2 were character height and font size. In HEIGHT2, we tested to see whether absolute difference (pixel difference) or percentage of difference would cause perceptual bias. Table A.11 shows that accuracy drops when the two factors disagree, and when percentage of difference decreases while pixel difference remains the same. This is evidence that percentage of difference is the real factor rather than pixel difference. We ran a two-way ANOVA test on percentage of font size difference and character height agreement. The results shown in table A.12 show that both factors are significant.

| pixelDiff | sizeDiff% | agree | neutral | disagree |
|---|---|---|---|---|
| 1 | 4.3% | 0.94 | 0.57 | 0.35 |
| 1 | 4.5% | 0.88 | 0.69 | 0.41 |
| 1 | 4.7% | 0.95 | 0.91 | 0.68 |
| 1 | 5% | 0.94 | 0.8 | 0.68 |
| 3 | 14% | 0.96 | 0.95 | 0.88 |
| 3 | 15% | 0.99 | 0.97 | 0.94 |

Table A.11: Breakdown Table for HEIGHT2

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiffPercent | 5 | 5 | 323 | 45.8820 | <.0001 |
| heightAgree | 2 | 2 | 323 | 83.8974 | <.0001 |
| sizeDiffPercent*adAgree | 10 | 10 | 323 | 11.1921 | <.0001 |

Table A.12: ANOVA Table for HEIGHT2

### A.2.3 Experiment HEIGHT3

The main factors in HEIGHT3 were the same as all of the other HEIGHT tests. In this test, we used a different font-family (the sans-serif font Roboto) to see if we would see the same effects. In table A.13, accuracy drops with the similar tendency in HEIGHT2 when font size difference is small and character heights disagree.

The ANOVA test results shown in table A.14 indicate that both font size difference and height agreement are significant factors in HEIGHT3.

| sizeDiff% | agree | neutral | disagree |
|---|---|---|---|
| 4.3% | 0.97 | 0.85 | 0.61 |
| 4.5% | 0.93 | 0.72 | 0.4 |
| 4.7% | 0.98 | 0.89 | 0.74 |
| 5% | 0.87 | 0.73 | 0.35 |
| 14% | 0.99 | 0.97 | 0.93 |
| 15% | 0.99 | 0.97 | 0.93 |

Table A.13: Breakdown Table for HEIGHT3

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiffPercent | 5 | 5 | 323 | 59.4186 | <.0001 |
| heightAgree | 2 | 2 | 323 | 36.0989 | <.0001 |
| sizeDiffPercent*adAgree | 10 | 10 | 323 | 12.5913 | <.0001 |

Table A.14: ANOVA Table for HEIGHT3

### A.2.4 Experiment HEIGHT4

The main factors of HEIGHT4 were character height and font size, as well. In HEIGHT4, we tested different font size baselines (20px, 30px) with the same percentage differences (3%,5%,10%,15%). In table A.15, results show that there was a greater effect of ascenders and descenders with a baseline of 30px than with a baseline of 20px.

In table A.16, ANOVA test results show that all three factors (font size difference, height agreement, and baseline) are significant factors.

| baseline | sizeDiff% | agree | neutral | disagree |
|---|---|---|---|---|
| 20 | 3% | 0.95 | 0.89 | 0.58 |
| 20 | 5% | 0.99 | 0.89 | 0.6 |
| 20 | 10% | 0.96 | 0.98 | 0.91 |
| 20 | 15% | 1 | 0.99 | 0.94 |
| 30 | 3% | 0.91 | 0.69 | 0.38 |
| 30 | 5% | 0.94 | 0.76 | 0.41 |
| 30 | 10% | 0.99 | 0.87 | 0.71 |
| 30 | 15% | 0.99 | 0.98 | 0.9 |

Table A.15: Breakdown Table for HEIGHT4

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiffPercent | 3 | 3 | 448 | 59.8059 | <.0001 |
| heightAgree | 2 | 2 | 448 | 88.3947 | <.0001 |
| sizeDiffPercent*adAgree | 6 | 6 | 448 | 16.5699 | <.0001 |
| baseline | 1 | 1 | 448 | 44.4990 | <.0001 |

Table A.16: ANOVA Table for HEIGHT4

### A.2.5 Experiment HEIGHT5

The main factors in HEIGHT5 were character height and font size. In HEIGHT5, we were looking for a "ceiling threshold" of font size difference beyond which character height would no longer have an effect. The results in table A.17 show that the ceilings for both baselines, 20px and 30px, are around 20% to 30%.

In table A.18, ANOVA test results again show that all three factors (font size difference, height agreement, and baseline) are significant factors for HEIGHT5. It is worth noting that since baseline was a between-subjects factor, there may be some interaction with differences across browsers for this factor.

| baseline | sizeDiff% | agree | neutral | disagree |
|---|---|---|---|---|
| 20 | 5% | 0.99 | 0.93 | 0.67 |
| 20 | 10% | 1 | 0.99 | 0.92 |
| 20 | 15% | 1 | 1 | 0.98 |
| 20 | 20% | 1 | 0.99 | 0.98 |
| 20 | 25% | 1 | 1 | 1 |
| 30 | 5% | 0.93 | 0.78 | 0.31 |
| 30 | 10% | 1 | 0.96 | 0.69 |
| 30 | 15% | 0.99 | 0.98 | 0.96 |
| 30 | 20% | 1 | 0.98 | 0.98 |
| 30 | 25% | 1 | 0.99 | 1 |

Table A.17: Breakdown Table for HEIGHT5

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiffPercent | 4 | 4 | 546 | 94.3882 | <.0001 |
| heightAgree | 2 | 2 | 546 | 207.2211 | <.0001 |
| sizeDiffPercent*adAgree | 8 | 8 | 546 | 37.7894 | <.0001 |
| baseline | 1 | 1 | 38 | 20.0918 | <.0001 |

Table A.18: ANOVA Table for HEIGHT5

## A.3 Width agreement experiments

### A.3.1 Experiment WIDTH1

The main factors in WIDTH1 were word width and font size. We varied the width and held the number of characters (word length) the same in WIDTH1. The results in table A.19 show that when width disagreed with font size, the accuracy dropped. In table A.20, ANOVA test results indicate that width difference and size difference cause significant changes in participant accuracy.

| | rawWidthDiff | |
|---|---|---|
| sizeDiff% | agree | disagree |
| 5% | 0.975 | 0.909 |
| 10% | 1.000 | 0.992 |
| 15% | 0.992 | 0.992 |
| 20% | 1.000 | 0.983 |

Table A.19: Breakdown Table for WIDTH1

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiff | 3 | 3 | 133 | 6.7717 | 0.0003 |
| rawWidthDiff | 1 | 1 | 133 | 11.3304 | 0.001 |
| sizeDiff*rawWidthDiff | 3 | 3 | 133 | 2.2867 | 0.0816 |

Table A.20: ANOVA Table for WIDTH1

### A.3.2 Experiment WIDTH2

The main factors in WIDTH2 were word length and font size. In WIDTH2, we controlled the width of the word and varied the number of characters. WIDTH2 is used to compare with the results of WIDTH1. The results of WIDTH2 in table A.21 show that length agreement did not change the accuracy much. Taking a further look in ANOVA test, shown in table A.22, there is no significant difference when word length changes, as well as size difference. Comparing WIDTH2 with WIDTH1 suggests that word width is the more important of the two co-varying factors.

| | lenDiff | |
|---|---|---|
| sizeDiff% | agree | disagree |
| 5% | 0.982 | 0.982 |
| 10% | 1.000 | 0.991 |
| 15% | 0.991 | 1.000 |
| 20% | 0.982 | 1.000 |

Table A.21: Breakdown Table for WIDTH2

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiff% | 3 | 3 | 126 | 1.466 | 0.227 |
| lenDiff | 1 | 1 | 126 | 0 | 1 |
| sizeDiff%*lenDiff | 3 | 3 | 126 | 1.3687 | 0.2554 |

Table A.22: ANOVA Table for WIDTH2

## A.4 Debiasing experiment

### A.4.1 Experiment BOX1

The main factors in BOX1 were word width and font size. From previous tests, we learned that width would have perceptual bias. In BOX1, we padded the word with rectangles to force each word had the same raw width (i.e., same width in font size 20px) and see if we can remove the biasing effect from word width. The results in table A.23 show that there is no obvious difference in factor agreement when the words were padded with rectangles. ANOVA test shown in table A.24 also indicates that word width is no longer a significant factor. However, font size difference is still a significant factor.

| | rawWidthDiff | | | |
|---|---|---|---|---|
| sizeDiff% | agree | neutral | disagree | AVG |
| 0.05 | 0.914 | 0.932 | 0.908 | 0.918 |
| 0.10 | 0.983 | 0.992 | 0.933 | 0.969 |
| 0.15 | 0.983 | 0.971 | 0.992 | 0.982 |
| 0.20 | 0.992 | 0.996 | 0.983 | 0.990 |
| AVG | 0.968 | 0.973 | 0.954 | 0.965 |

Table A.23: Breakdown Table for BOX1

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| sizeDiff% | 3 | 3 | 209 | 10.8814 | <.0001 |
| rawWidthDiff | 2 | 2 | 209 | 0.5164 | 0.5974 |
| sizeDiff%*rawWidthDiff | 6 | 6 | 209 | 1.07 | 0.3816 |

Table A.24: ANOVA Table for BOX1

## A.5 Alternate experiment

### A.5.1 Experiment BIG1

The main factors in BIG1 were font size, number of near misses, and word length agreement. From previous tests, we learned that font size and length agreement were significant when comparing two highlighted words. In BIG1, we tested to see if this effect can be seen using a different task: selecting the *largest* word rather than comparing two highlighted words. We also tested a factor that we called "near misses". Near misses are words that are close to the largest font size, but not quite. We tested to see whether the number of near misses would have an effect on user ability to find the biggest word. The results in table A.25 show that we see a similar bias when having participants pick the biggest word as we did when they were comparing two words. The results also show that instances with more near misses would have worse accuracy. The ANOVA test shown in table A.26 also indicates that these three main factors are significant.

| sizeDiff% | #NearMisses | agree | neutral | disagree |
|---|---|---|---|---|
| 5% | 1 | 0.95 | 0.91 | 0.75 |
| 10% | 1 | 0.97 | 0.98 | 0.9 |
| 15% | 1 | 0.99 | 0.97 | 0.91 |
| 20% | 1 | 0.99 | 0.99 | 0.93 |
| 5% | 4 | 0.83 | 0.74 | 0.57 |
| 10% | 4 | 0.91 | 0.91 | 0.77 |
| 15% | 4 | 0.96 | 0.97 | 0.97 |
| 20% | 4 | 1 | 0.99 | 1 |

Table A.25: Breakdown Table for BIG1

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| nm | 1 | 1 | 414 | 33.6625 | <.0001 |
| sizeDiff | 3 | 3 | 414 | 5.8224 | 0.0007 |
| nm*sizeDiff | 3 | 3 | 414 | 10.1282 | <.0001 |
| lengthDiff | 2 | 2 | 414 | 10.0968 | <.0001 |
| nm*lengthDiff | 2 | 2 | 414 | 0.5275 | 0.5904 |
| sizeDiff*lengthDiff | 6 | 6 | 414 | 1.0411 | 0.3979 |
| nm*sizeDiff*lengthDiff | 6 | 6 | 414 | 0.8194 | 0.5553 |

Table A.26: ANOVA Table for BIG1

### A.5.2 Experiment BIG2

The main factors in BIG2 were font size, number of near misses, and length difference. In BIG2, we looked at more variations of word length to see if larger degrees of disagreement would have more of an effect. The results in table A.27 show that larger word length differences had larger drops of accuracy when factor agreements changed. The ANOVA test shown in table A.28 also indicates that length difference were significant.

| sizeDiff% | #NearMisses | lenDiff | agree | disagree |
|---|---|---|---|---|
| 5% | 1 | 1 | 0.95 | 0.71 |
| 10% | 1 | 1 | 0.95 | 0.95 |
| 15% | 1 | 1 | 0.93 | 0.83 |
| 20% | 1 | 1 | 1 | 0.97 |
| 5% | 1 | 3 | 0.86 | 0.61 |
| 10% | 1 | 3 | 0.95 | 0.71 |
| 15% | 1 | 3 | 1 | 0.8 |
| 20% | 1 | 3 | 1 | 0.85 |
| 5% | 1 | 5 | 0.86 | 0.6 |
| 10% | 1 | 5 | 0.97 | 0.71 |
| 15% | 1 | 5 | 0.98 | 0.71 |
| 20% | 1 | 5 | 0.97 | 0.83 |
| 5% | 4 | 1 | 0.74 | 0.54 |
| 10% | 4 | 1 | 0.83 | 0.73 |
| 15% | 4 | 1 | 0.88 | 0.84 |
| 20% | 4 | 1 | 0.95 | 0.95 |
| 5% | 4 | 3 | 0.71 | 0.5 |
| 10% | 4 | 3 | 0.9 | 0.6 |
| 15% | 4 | 3 | 0.91 | 0.72 |
| 20% | 4 | 3 | 0.97 | 0.9 |
| 5% | 4 | 5 | 0.74 | 0.4 |
| 10% | 4 | 5 | 0.93 | 0.56 |
| 15% | 4 | 5 | 0.97 | 0.7 |
| 20% | 4 | 5 | 0.98 | 0.78 |

Table A.27: Breakdown Table for BIG2

| Source | Nparm | DF | DFDen | F Ratio | Prob >F |
|---|---|---|---|---|---|
| lenDiff | 5 | 5 | 846 | 7.9977 | <.0001 |
| sizeDiff | 3 | 3 | 846 | 3.0159 | 0.0292 |
| lenDiff*sizeDiff | 15 | 15 | 846 | 0.9649 | 0.4909 |
| nm | 1 | 1 | 846 | 7.0023 | 0.0083 |
| lenDiff*nm | 5 | 5 | 846 | 0.2384 | 0.9455 |
| sizeDiff*nm | 3 | 3 | 846 | 1.3553 | 0.2552 |
| lenDiff*sizeDiff*nm | 15 | 15 | 846 | 0.4299 | 0.9707 |

Table A.28: ANOVA Table for BIG2

## REFERENCES

Afzal, Shehzad, Ross Maciejewski, Yun Jang, Niklas Elmqvist, and David S Ebert. 2012. Spatial text visualization using automatic typographic maps. *IEEE Transactions on Visualization & Computer Graphics* 18(12):2556–2564.

Albers, Danielle, Colin Dewey, and Michael Gleicher. 2011. Sequence surveyor: Leveraging overview for scalable genomic alignment visualization. *Visualization and Computer Graphics, IEEE Transactions on* 17(12):2392–2401.

Alexander, Eric, and Michael Gleicher. 2015. Task-driven comparison of topic models. *IEEE Transactions on Visualization and Computer Graphics*.

———. 2016. Assessing topic representations for gist-forming. In *Proceedings of the international working conference on advanced visual interfaces*. ACM. In press.

Alexander, Eric, Joe Kohlmann, Robin Valenza, Michael Witmore, and Michael Gleicher. 2014. Serendip: Topic model-driven visual exploration of text corpora. In *Visual analytics science and technology (vast), 2014 ieee conference on*, 173–182. IEEE.

Alper, Basak, Huahai Yang, Eben Haber, and Eser Kandogan. 2011. Opinionblocks: Visualizing consumer reviews. In *Proc. of the ieee workshop on interactive visual text analytics for decision making*.

AlSumait, Loulwah, Daniel Barbará, James Gentle, and Carlotta Domeniconi. 2009. Topic significance ranking of lda generative models. In *Machine learning and knowledge discovery in databases*, 67–82. Springer.

Amershi, Saleema, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd annual acm conference on human factors in computing systems*, 337–346. ACM.

Andrzejewski, David, Xiaojin Zhu, Mark Craven, and Benjamin Recht. 2011. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *Ijcai proceedings-international joint conference on artificial intelligence*, vol. 22, 1171.

Baron, Alistair, Paul Rayson, and Dawn Archer. 2011. Quantifying early modern english spelling variation: change over time and genre. In *Conf. new methods in historical corpora*.

Bateman, Scott, Carl Gutwin, and Miguel Nacenta. 2008. Seeing things in the clouds: The effect of visual features on tag cloud selections. In *Proceedings of the nineteenth acm conference on hypertext and hypermedia*, 193–202. ACM.

Bertin, Jacques. 2011. *Semiology of graphics: diagrams, networks, maps*. Esri Press.

Blei, D. 2012. Probabalistic topic models. *Communications of the ACM* 55(4): 77–84.

Blei, D.M., and J.D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on machine learning*, 113–120. ACM.

Blei, D.M., A.Y. Ng, and M.I. Jordan. 2003. Latent dirichlet allocation. *J. Machine Learning Research* 3:993–1022.

Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer. 2011. $D^3$ data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on* 17(12): 2301–2309.

Brath, Richard, and Ebad Banissi. 2015. Evaluating lossiness and fidelity in information visualization. In *Is&t/spie electronic imaging*, 93970H–93970H. International Society for Optics and Photonics.

Brewer, Cynthia A, Geoffrey W Hatchard, and Mark A Harrower. 2003. Colorbrewer in print: A catalog of color schemes for maps. *Cartography and Geographic Information Science* 30(1):5–32.

Brown, Eli T., Jingjing Liu, Carla E. Brodley, and Remco Chang. 2012. Disfunction: Learning distance functions interactively. In *Ieee visual analytics science and technology*, 83–92. IEEE.

Chaney, A.J.B., and D.M. Blei. 2012. Visualizing topic models. In *Proc. aaai on weblogs and social media*.

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

Choo, Jaegul, Changhyun Lee, Chandan K Reddy, and Haesun Park. 2013. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *Visualization and Computer Graphics, IEEE Transactions on* 19(12): 1992–2001.

Chuang, J., C.D. Manning, and J. Heer. 2012a. Termite: visualization techniques for assessing textual topic models. In *Proc. advanced visual interfaces*, 74–77. ACM.

Chuang, J., D. Ramage, C. Manning, and J. Heer. 2012b. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proc. acm human factors in computing systems*, 443–452. ACM.

Chuang, Jason, Sonal Gupta, Christopher Manning, and Jeffrey Heer. 2013. Topic model diagnostics: Assessing domain relevance via topical alignment. In *Proceedings of the 30th international conference on machine learning (icml-13)*, 612–620.

Clement, Tanya, Catherine Plaisant, and Romain Vuillemot. 2009. The story of one: Humanity scholarship with visualization and text analysis. *Relation* 10(1.43):8485.

Cleveland, William S, and Robert McGill. 1984. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association* 79(387):531–554.

Climer, Sharlee, and Weixiong Zhang. 2006. Rearrangement clustering: Pitfalls, remedies, and applications. *J. Machine Learning Research* 7:919–943.

Collins, Christopher, Sheelagh Carpendale, and Gerald Penn. 2009a. Docuburst: Visualizing document content using language structure. In *Computer graphics forum*, vol. 28, 1039–1046. Wiley Online Library.

Collins, Christopher, Fernanda B. Viégas, and Martin Wattenberg. 2009b. Parallel tag clouds to explore and analyze faceted text corpora. In *Proc. of the ieee symp. on visual analytics science and technology (vast)*.

Collins, Jeff, and Dave Kaufer. 2001. Description of docuscope.

Correll, Michael, Eric Alexander, Danielle Albers, Alper Sarikaya, and Michael Gleicher. 2014. Navigating reductionism and holism in evaluation. In *Beliv*

*'14 proceedings of the fifth workshop on beyond time and errors: Novel evaluation methods for visualization*, 23–26.

Correll, Michael, Eric Alexander, and Michael Gleicher. 2013. Quantity estimation in visualizations of tagged text. In *Proc. acm human factors in computing systems*. ACM.

Correll, Michael, and Michael Gleicher. 2012. What Shakespeare taught us about text visualization. In *Ieee visualization workshop proceedings, interactive visual text analytics*.

———. 2014. Error bars considered harmful: Exploring alternate encodings for mean and error. *IEEE Transactions on Visualization and Computer Graphics* 20(12):2142–2151. IEEE Vis Conference, InfoVis track, to appear.

———. 2016. The semantics of sketch: A visual query system for time series data. In *Proceedings of the 2016 ieee conference on visual analytics science and technology (vast)*. IEEE. To appear.

Correll, Michael, Michael Witmore, and Michael Gleicher. 2011. Exploring Collections of Tagged Text for Literary Scholarship. *Computer Graphics Forum* 30(3): 731–740.

Crossno, Patricia J, Andrew T Wilson, Timothy M Shead, and Daniel M Dunlavy. 2011. Topicview: Visually comparing topic models of text collections. In *Tools with artificial intelligence (ictai), 2011 23rd ieee international conference on*, 936–943. IEEE.

Cui, Weiwei, Shixia Liu, Li Tan, Conglei Shi, Yangqiu Song, Zekai Gao, Huamin Qu, and Xin Tong. 2011. Textflow: Towards better understanding of evolving topics in text. *IEEE TVCG* 17(12):2412–2421.

Davies, Jason. 2015. d3-cloud. `https://github.com/jasondavies/d3-cloud`.

Davies, Mark. 2011. Word frequency data from the corpus of contemporary american english (COCA). `http://www.wordfrequency.info`.

DiMaggio, Paul, Manish Nag, and David Blei. 2013. Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of us government arts funding. *Poetics* 41(6):570–606.

Dou, Wenwen, Xiaoyu Wang, Remco Chang, and William Ribarsky. 2011. ParallelTopics: A probabilistic approach to exploring document collections. In *Ieee visual analytics science and technology*, 231–240. IEEE.

Dou, Wenwen, Li Yu, Xiaoyu Wang, Zhiqiang Ma, and William Ribarsky. 2013. HierarchicalTopics: Visually exploring large text collections using topic hierarchies. *IEEE TVCG* 19(12):2002–2011.

Endert, Alex, Patrick Fiaux, and Chris North. 2012a. Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *Visualization and Computer Graphics, IEEE Transactions on* 18(12):2879–2888.

———. 2012b. Semantic interaction for visual text analytics. In *Proc. acm human factors in computing systems*, 473–482. ACM.

Endert, Alex, Chao Han, Dipayan Maiti, Leanna House, Scotland Leman, and Chris North. 2011. Observation-level interaction with statistical models for visual analytics. In *Ieee visual analytics science and technology*, 121–130.

Filippova, Darya, Aashish Gadani, and Carl Kingsford. 2012. Coral: an integrated suite of visualizations for comparing clusterings. *BMC bioinformatics* 13(1):276.

van der Geest, Thea, and Raymond van Dongelen. 2009. What is beautiful is useful-visual appeal and expected information quality. In *Professional communication conference, 2009. ipcc 2009. ieee international*, 1–5. IEEE.

Gleicher, Michael. 2013. Explainers: expert explorations with crafted projections. *IEEE TVCG* 19(12):2042–51.

Griffiths, DMBTL, and MIJJB Tenenbaum. 2004. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems* 16:17.

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

Haber, Lyn R, Ralph Norman Haber, and Karen R Furlin. 1983. Word length and word shape as sources of information in reading. *Reading Research Quarterly* 165–189.

Haberman, Jason, and David Whitney. 2012. Ensemble perception: Summarizing the scene and broadening the limits of visual processing. *From perception to consciousness: Searching with Anne Treisman* 339–349.

Halvey, Martin J, and Mark T Keane. 2007. An assessment of tag presentation techniques. In *Proceedings of the 16th international conference on world wide web*, 1313–1314. ACM.

Harris, Jacob. 2011. Word clouds considered harmful, blog, http://www.niemanlab.org/2011/10/word-clouds-considered-harmful/.

Havre, Susan, Beth Hetzler, and Lucy Nowell. 2000. ThemeRiver: Visualizing theme changes over time. In *Proc. ieee information visualization*.

Hearst, Marti A, and Daniela Rosner. 2008. Tag clouds: Data analysis tool or social signaller? In *Hawaii international conference on system sciences, proceedings of the 41st annual*, 160–160. IEEE.

Henry, Nathalie, and Jean-Daniel Fekete. 2007. MatLink: Enhanced matrix visualization for analyzing social networks. *Human-Computer Interaction–INTERACT 2007* 288–302.

Hofmann, Thomas. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval*, 50–57. ACM.

Hope, Jonathan, and Michael Witmore. 2010. The hundredth psalm to the tune of "green sleeves": Digital approaches to shakespeare's language of genre. *Shakespeare Quarterly* 61(3):357–390.

Hu, Yuheng, Ajita John, Fei Wang, and Subbarao Kambhampati. 2012. Et-lda: Joint topic modeling for aligning events and their twitter feedback. In *Aaai*, vol. 12, 59–65.

Jänicke, Stefan, Greta Franzini, M Cheema, and Gerik Scheuermann. 2015. On close and distant reading in digital humanities: A survey and future challenges. *Proc. of EuroVisâĽ"STARs* 83–103.

Jockers, Matthew L. 2013. *Macroanalysis: Digital methods and literary history*. University of Illinois Press.

Joia, Paulo, Fernando V Paulovich, Danilo Coimbra, José Alberto Cuminato, and Luis Gustavo Nonato. 2011. Local Affine Multidimensional Projection. *IEEE TVCG* 17(12):2563–2571.

Keim, Daniel A., and Daniela Oelke. 2007. Literature fingerprinting: A new method for visual literary analysis. In *Ieee visual analytics science and technology*, 115–122. IEEE.

Koch, Steffen, Markus John, Michael Wörner, Andreas Müller, and Thomas Ertl. 2014. VarifocalreaderâŁ"in-depth visual analysis of large text documents. *IEEE transactions on visualization and computer graphics* 20(12):1723–1732.

Kullback, Solomon, and Richard A Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics* 79–86.

Kuo, Byron YL, Thomas Hentrich, Benjamin M Good, and Mark D Wilkinson. 2007. Tag clouds for summarizing web search results. In *Proceedings of the 16th international conference on world wide web*, 1203–1204. ACM.

Lancichinetti, Andrea, M Irmak Sirer, Jane X Wang, Daniel Acuna, Konrad Körding, and Luís A Nunes Amaral. 2015. High-reproducibility and high-accuracy method for automated topic classification. *Physical Review X* 5(1):011007.

Lee, Bongshin, Mary Czerwinski, George Robertson, and Benjamin B Bederson. 2005. Understanding research trends in conferences using paperlens. In *Chi'05 extended abstracts on human factors in computing systems*, 1969–1972. ACM.

Lohmann, Steffen, Jürgen Ziegler, and Lena Tetzlaff. 2009. Comparison of tag cloud layouts: Task-related performance and visual exploration. In *Human-computer interaction–interact 2009*, 392–404. Springer.

Maps, Axis. 2015. Typographic maps. `http://www.axismaps.com/`.

McCallum, Andrew Kachites. 2002. Mallet: A machine learning for language toolkit. Http://mallet.cs.umass.edu.

Meeks, Elijah. 2012. Using word clouds for topic modeling results, blog, https://dhs.stanford.edu/algorithmic-literacy/using-word-clouds-for-topic-modeling-results/.

Mimno, David, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing*, 262–272. Association for Computational Linguistics.

Moretti, Franco. 2005. *Graphs, maps, trees: Abstract models for a literary history*. Verso Books.

Mueller, C., B. Martin, and A. Lumsdaine. 2007. A comparison of vertex ordering algorithms for large graph visualization. In *2007 6th international asia-pacific symposium on visualization*, 141–148. IEEE.

Muhlbacher, Thomas, Harald Piringer, Samuel Gratzl, Michael Sedlmair, and Marc Streit. 2014. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *Visualization and Computer Graphics, IEEE Transactions on* 20(12):1643–1652.

Nacenta, Miguel, Uta Hinrichs, and Sheelagh Carpendale. 2012. Fatfonts: combining the symbolic and visual aspects of numbers. In *Proceedings of the international working conference on advanced visual interfaces*, 407–414. ACM.

Newman, David, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on digital libraries*, 215–224. ACM.

Ongaro, Luca. 2014. jQCloud: jQuery plugin for drawing neat word clouds that actually look like clouds. `https://github.com/lucaong/jQCloud`.

Paulovich, F.V., and R. Minghim. 2006. Text map explorer: a tool to create and explore document maps. In *Conf. information visualisation*, 245–251. IEEE.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *J. Machine Learning Research* 12:2825–2830.

Plaisant, Catherine, James Rose, Bei Yu, Loretta Auvil, Matthew G. Kirschenbaum, Martha Nell Smith, Tanya Clement, and Greg Lord. 2006. Exploring erotics in Emily Dickinson's correspondence with text mining and visual interfaces. In *Proc. acm/ieee joint conf. digital libraries*, 141–150. ACM Press.

Řehůřek, Radim, and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA. http://is.muni.cz/publication/884893/en.

Rivadeneira, Anna W, Daniel M Gruen, Michael J Muller, and David R Millen. 2007. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the sigchi conference on human factors in computing systems*, 995–998. ACM.

Rohrer, R.M., D.S. Ebert, and J.L. Sibert. 1998. The shape of Shakespeare: visualizing text using implicit surfaces. In *Proc. ieee information visualization*, 121–129. IEEE.

Ross, John, and David C Burr. 2010. Vision senses number directly. *Journal of Vision* 10(2):10.

Sandhaus, Evan. 2008. The New York Times Annotated Corpus LDC2008T19. DVD. Philadelphia: Linguistic Data Consortium.

Siirtola, Harri. 1999. Interaction with the reorderable matrix. In *Proc. ieee information visualization*, 272–277. IEEE.

Skupin, André. 2004. The world of geography: Visualizing a knowledge domain with cartographic means. *Proceedings of the National Academy of Sciences* 101(suppl 1):5274–5278.

Stasko, John, Carsten Görg, and Zhicheng Liu. 2008. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization* 7(2):118–132.

Stevens, Keith, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 952–961. Association for Computational Linguistics.

Szafir, Danielle Albers, Steve Haroz, Michael Gleicher, and Steven Franconeri. 2016. Four types of ensemble coding in data visualizations. *Journal of vision* 16(5):11–11.

Thudt, Alice, Uta Hinrichs, and Sheelagh Carpendale. 2012. The bohemian bookshelf: supporting serendipitous book discoveries through information visualization. In *Proc. acm human factors in computing systems*, 1461–1470. ACM.

Torget, Andrew J, Rada Mihalcea, Jon Christensen, and Geoff McGhee. 2011. Mapping texts: Combining text-mining and geo-visualization to unlock the research potential of historical newspapers.

Trattner, Christoph, Denis Helic, and Markus Strohmaier. 2014. Tag clouds. In *Encyclopedia of social network analysis and mining*, 2103–2107. Springer.

Viégas, Fernanda B, and Martin Wattenberg. 2008. Timelines tag clouds and the case for vernacular visualization. *interactions* 15(4):49–52.

Viégas, Fernanda B, Martin Wattenberg, and Jonathan Feinberg. 2009. Participatory visualization with wordle. *Visualization and Computer Graphics, IEEE Transactions on* 15(6):1137–1144.

Wallach, Hanna M, David Minmo, and Andrew McCallum. 2009a. Rethinking lda: Why priors matter.

Wallach, Hanna M, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009b. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, 1105–1112. ACM.

Wang, Chong, David Blei, and David Heckerman. 2012. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*.

Wattenberg, Martin, and Fernanda B Viégas. 2008. The word tree, an interactive visual concordance. *Visualization and Computer Graphics, IEEE Transactions on* 14(6):1221–1228.

Wei, Furu, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. 2010. Tiara: a visual exploratory text analytic system. In *Proc. acm knowledge discovery and data mining*, 153–162. ACM.

Xu, Panpan, Yingcai Wu, Enxun Wei, Tai-Quan Peng, Shixia Liu, Jonathan JH Zhu, and Huamin Qu. 2013. Visual analysis of topic competition on social media. *Visualization and Computer Graphics, IEEE Transactions on* 19(12):2012–2021.

Yao, Limin, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining*, 937–946. ACM.

Zhao, Wayne Xin, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *European conference on information retrieval*, 338–349. Springer.

Zimek, Arthur, Erich Schubert, and Hans-Peter Kriegel. 2012. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining* 5(5):363–387.