DocuVis: Interactive Document Clustering and Visualization with Latent Dirichlet Allocation

Alan Peral and Ya Xu

Abstract—Proper clustering and visualization tools simplify the process of information retrieval, navigation, and organization when dealing with a variety of documents. We present DocuVis, an interactive visualization system for document clustering and organization. We utilize a force-directed graph to visualize the topic clusters based on the Latent Dirichlet Allocation (LDA) topic model analysis and the D3 visualization package. We incorporate a variety of visualization and navigation tools to provide users with information about document sets they provide, helping people organize their files easily and automatically. We also demonstrate the effectivity of the DocuVis platform in integrating into existing research-oriented workflows.

Index Terms—Document Clustering, Latent Dirichlet Allocation, force-directed graph, data visualization, InfoVis.

1 Introduction

The automatic organization of documents into related clusters has been showed to aid in the information retrieval process [11]. The clustering problem, or the attempt to find similarities between groups of data with the use of a similarity function, has been widely studied in a variety of applications [7]. Accordingly, there have been a multitude of proposed clustering algorithms that are suited for differing tasks, making the selection of a clustering algorithm for general use quite the challenging task. In particular, we are interested in the application of clustering towards document organization and browsing, corpus summarization, and document classification. Typically, for these tasks naive techniques such as the k-means algorithm, certain hierarchical algorithms, or quantitative data clustering algorithms used with the frequencies of words in a collection do not work well for clustering text data [1]. There are a number of issues that necessitate more sophisticated algorithms:

- The dimensionality of text representation is large, but the underlying data is sparse.
- The lexicon of a given corpus of documents may be large, but the words are typically correlated with one another, which means that the number of concepts in the data is smaller than the feature space.
- The number of words in the differing documents vary widely. Therefore, normalizing document representations is an important aspect of the clustering problem.

This paper thus suggests an approach to document and clustering based on Latent Dirichlet Allocation (LDA) and self-directed force graphs that addresses some of these concerns. LDA is a probabilistic topic model that can determine a topic distribution from the word content of each document in a collection. In fact, LDA is a fully generative approach to language modeling that overcomes inconsistencies of generative semantics produced by Probabilistic Latent Semantics Indexing (PLSI) [5]. Compared to other topic models, LDA models strike a balance in common errors that occur, like the limitation of document sizes (common in naive Bayes models and Gaussian mixtures models) or overfitting/underfitting issues (common in PLSI models).

For the visualization tools, we utilize D3, a javascript library well-suited for scientific visualization, to draw a force-directed layout graph. Force-directed layout graphs are drawn by a force-directed algorithm, which is usually a simulation of physics force-based layout systems. The original system works as a simulation of physics systems like molecular mechanics, and typically these systems combine attractive forces between adjacent vertices with repulsive forces between all pairs of vertices, in order to seek a layout in which edge lengths are small while vertices are well-separated. We build several navigation and informational tools on top of this library to create an organizational tool that can help people group together and organize large collections of documents. We include the ability to retrieve automatically generated keywords and summaries from any given documents using the TextRank algorithm, a lightweight and efficient solution that is suitable for web applications.

2 Motivation and Objective

Users today are overloaded with information via the web and other media. With so much written content being generated daily (and without even considering other media), attempting to keep up-to-date and organized with the diverse array of information being produced becomes an almost impossible task. This makes it increasingly attractive to find ways to easily keep textual information and documents accessible, organized, and digestible to make them easier to find and navigate through. Organizing data into groups is one of the most fundamental ways of learning and understanding, and as such, much research has been conducted into how to properly and understandably generate sensible groupings from document sets [7]. There is a challenge not only in using topic modeling algorithms that are robust

• *Alan Peral is with the University of California, Santa Cruz. Email: aperalor@ucsc.edu*

Manuscript received Feb 1, 2019; revised Feb 1, 2019.

and flexible, but also in generating visualizations that are easy to use and informative, so that any potential user can make use of the system without any training or difficulty. It becomes an interesting problem then, to decipher how to provide as much useful information without cluttering the visualization beyond comfort for users.

For our project, we have chosen to convey limited amounts of data. We convey the topic models generated by the LDA algorithm, as well as the distribution of topics within a document (as one document may contain more than one of the proposed topics). We also show the name of every document on, for ease of navigation, as well as the keywords corresponding to each document, and an automatically generated extractive summary to accompany and aid in understanding of said document. Extractive methods for summarization focus on identifying and selecting important sentences in the text (verbatim). Thus, while the summaries we provide contain the most pertinent information found in a document, the produced summary always runs the risk of sounding choppy as the most important sentences in a document may not necessarily be immediately related.

2.1 Related Works

Topic-based text visualization seeks to identify and explore topics (clusters) conveyed by a given set of documents. Traditional methods for document clustering include naive Bayes, maximum entropy, and support vector machine techniques. The idea behind these approaches is to convert each document into a vector inside the hyperspace and then calculate the distance between these vectors, using it to show the dissimilarity between documents. Thus, document clustering is formulated as a mathematical grouping of vectors [9].

Visualization is a useful tool for understanding complex and high dimensional data, and it enables us to browse intuitively through huge amounts of data. Previous papers have proposed a variety of different document visualization methods [6].

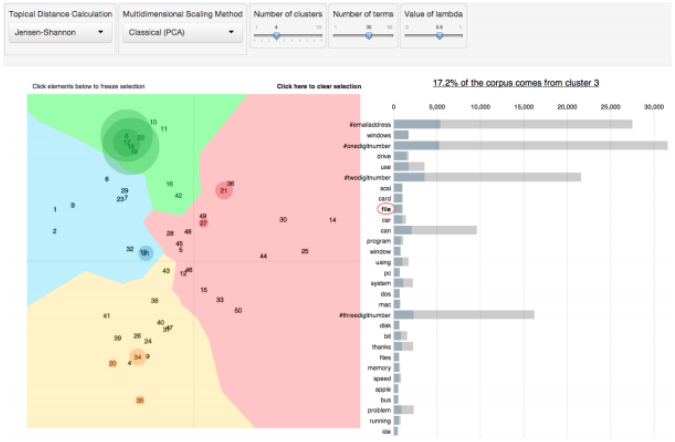

Fig. 1. One of the views from LDAvis displaying the segmenting of topics into four clusters.

Other researchers have visualized the results of the LDA topic modeling as well. One of the most widely used LDA visualizations is LDAvis (seen in Figure 1), a web-based interactive visualization of topics that was built using a combination of R and D3. LDAvis primarily allows users to select a topic to reveal the most relevant terms for the topic, as well as the ability to select a term to reveal its conditional distribution over topics [13]. These two primary functions allow users to better understand the topic-term relationships in a fitted LDA model. Another LDA oriented visualization is LDAExplore, which is centered on document visualization using LDA topic modeling. LDAExplore employs a variety of views to visualize topic and word distributions generated from the document corpus, allowing users to interact with them [4]. Another system that focuses on LDA topic modeling is called iVisClustering, and is in fact similar to the work presented in this paper. iVisClustering graphs documents based on topic similarity, with nodes representing documents, colors representing topics (where documents categorized under the same topic share a colour), and edges represent how similar the documents are based on cosine similarity [8]. The system provides many different views and interactions so that the user may have some influence over the topic modeling system, but looking at Figure 2 we see that to truly be able to use the iVisClustering system adequately, some sort of in-depth training or tutorial would be necessary. This is because it has a many different views that portray information in vastly different ways. Figure 2 shows 7 different views that a user would have to interact with and understand in order to fully utilize the power of iVisClustering. Some views, like the one labeled F, are not immediately understandable and would require a thorough explanation for any user. In other words, while iVisClustering is a powerful and informative system, it lacks the accessibility for a wide audience needed for it to truly be useful. However, view A is a very similar view to the one presented in this work, as it is also based on a force-directed graph representation of document sets.

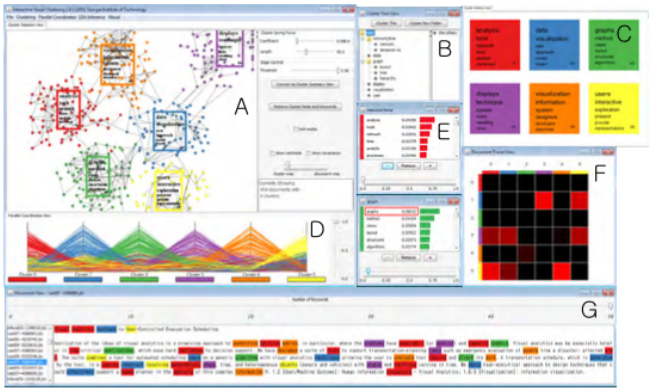

Fig. 2. The many different views of the iVisClustering system.

2.2 Text Summarization

The subfield of automatic text summarization has been the subject of investigation by the Natural Language Processing (NLP) community for the last half century. Early approaches in the 50s and 60s focused on extracting important sentences through the use of features like *word* and *phrase*

frequency, position in the text, and *key phrases*. Later work in the 90s focused on extracting important sentences with the use of a naive-Bayes methods, which had the particular flaw of assuming feature independence [3]. A variety of other models have been used since then, including decision trees, hidden markov models, and log linear models, but surprisingly could not improve on a strong baseline set in 2002 by the DUC with statistical significance [1]. This baseline corresponded to choosing the first n sentences in an article, which highlights the journalistic tendency to put the pertinent information of an article at the beginning [3].

In 2004, the TextRank algorithm was developed on top of Google's famous PageRank graph ranking algorithm and was able to successfully extract text summarizations [10]. TextRank was based in the representation of text as a graph, and at the time of release improved upon preexisting baselines set by DUC [10]. One of its many benefits was that it didn't need preexisting data to be trained because of its graph approach to text summarization. Much like us humans, TextRank relies only on a document to generate its summary, unlike other methods that require training corpora. Since its development, TextRank has proved to be an excellent approach to fast and accessible text summarization. Additionally, TextRank's language and domain independence have made it an attractive option for other researchers to improve and build upon [2]. For our purposes, we simply utilize the TextRank algorithm as described in its original paper.

3 VISUALIZATION

We use the *gensim* Python library to implement the LDA topic model in our visualization system due to its exceptional ability to handle Natural Language Processing problems. The output of the LDA topic model consists of a list of documents and the weight of the topics contained within them, as well as the topics and the combination of weights and keywords that contribute to each topic. Thus, the output of an arbitrary model with n documents and m topics in the set may look like this:

$$
\begin{aligned}
&document_0 = topic_0 \times weight_0 + topic_1 \times weight_1 + \dots \\
&document_1 = topic_0 \times weight_0 + topic_1 \times weight_1 + \dots \\
&\vdots \\
&document_n = topic_0 \times weight_0 + topic_1 \times weight_1 + \dots \\
&topic_0 = \text{“}word_1\text{”} \times weight_1 + \text{“}word_2\text{”} \times weight_2 + \dots \\
&\vdots \\
&topic_m = \text{“}word_3\text{”} \times weight_3 + \text{“}word_1\text{”} \times weight_1 + \dots
\end{aligned}
$$

Note that some documents may only consist of one document (i.e., document 1 may have a topic 2 weight of 1.0), and that topics may consist of completely different words.

1. https://www-nlpir.nist.gov/projects/duc/

This LDA topic model output then serves as the base upon which we design our visualization. We utilize a force-directed layout algorithm to plot the graph, which utilizes the attractive and repulsive forces between nodes to produce a stable configuration for the graph. This stable configuration draws groups similar nodes together while attempting to maintain ample spacing between different topics and nodes, rendering a readable and aesthetically pleasing graph. For our purposes, the force-directed algorithm is especially well suited to displaying the inter-document relations and document-topic relations as the repulsive and attractive forces within the simulation. However, the LDA topic model output does not align with the input for the force-directed algorithm, and so some additional preprocessing on the LDA topic model output data is necessary as an intermediary step before the force-directed algorithm implemented in D3 can be used. The two main components that are needed for the force-directed algorithm are nodes, represented in our case by the document names as well as the topic name (cluster) nodes, and the links, which exist between documents in the same cluster (inner links), as well as between the same document when it exists in multiple clusters (outer links). For each link, the weight of the force simulated is calculated as the cosine distance between different nodes, with the vectors that are used for this calculation being the topic array for each node. This is the output of the LDA topic model. For the nodes that just contain the topic name, we simply assume that they consist of a single topic.

3.1 Multi-level Force Directed Graph

In traditional applications of the force-directed algorithm, nodes tend to represent single objects, given that the projections from objects to nodes are unique. However, we did not wish to limit the document categorization process to one topic per document, as we understood that a significant amount of data could be lost by enforcing this policy. Consequently, we had to make it possible for documents to appear multiple times in our visualization, in multiple clusters if necessary. So if a document was best encapsulated by a split of two separate topics, then the document should appear in the corresponding cluster for each topic, rather than being limited to a single projection for each of the nodes in a typical force directed projection.

This is where a multi-level approach comes in handy. In particular, we design a two-level force directed graph to generate the appropriate behaviour for our visualization. The first level is at the inner cluster level, for which we draw a normal force-directed graph for each individual topic cluster. The cluster consists of nodes that share the same group feature, which will be document nodes that have the same topic in addition to the singular informative topic name node. This inner cluster level consists purely of inner links since all the nodes are in the same group. The second level operates on the inter-cluster level, and is simulated in relation to the outer links defined earlier. This two-level structure guarantees that a single document can have projections in different clusters, and will allow the position of different nodes in the same cluster to represent their relations with each other while still conveying the

relations amongst different topic clusters based on their positions.

3.2 Phrase Detection

In order to decide which tokens to use as candidates for the topic analysis, we construct a dictionary to include all possible tokens for the topic content. Usually this is usually a set of single words but we would want to include n-grams as well. However, not all two words that are close to each other can be counted as a phrase since most word pairs might be meaningless. We focus on phrase detection based on frequency in order to detect meaningful phrases automatically. If a phrase appears more frequently than a value we set, we can deem it to be meaningful and incorporate it as part of the dictionary. This is helpful because we do not need any prior knowledge of our metadata, but in some cases there exists the possibility that we might miss on some potential meaningful phrase if their frequency happens to be too low.

3.3 Optimal Topic Number

For the LDA topic model, the number of topics is a parameter that has to be supplied before fitting the model. The number of topics suggested will influence the model, so it is essential to find the right number of topics to best encapsulate the spread of the document set. Setting the number too low might miss out on meaningful topics, while setting it too high might make certain topics indistinguishable from each other. We thus seek to alleviate this issue by incorporating a measurement called topic coherence to detect how good a topic model is [12]. Thus, by running the LDA model with a different range for the number of topics to generate, and then measuring the topic coherence afterwards, we can ascertain with certainty which number of topics is best for a particular document set.

3.4 Additional Features

Topic Cluster Boundary: Rather than drawing the cluster boundary for each topic as a circle with a fixed center at the topic node, we instead draw the smallest circle that contains all nodes in a single topic cluster. This is done by implementing Welzl's algorithm which computes the minimal enclosing circle [14]. This makes for a more aesthetically pleasing clustering appearance.

Color: In order to make the different topic groupings easily distinguishable, we add colour to indicate which topic group a set of document nodes falls under. Referring to Figure 4, we note that the mentoring topic group is encoded in yellow, the graduate group encoded in green, the faculty group is in red, and the investigator group is in blue. This makes it immediately apparent that there is a distinction between the groupings of nodes under each topic.

Mouse hover elements: When a user hovers their mouse cursor over the circle 'document' nodes, a tooltip will appear above the circle nodes to show the document's name, and the corresponding nodes will increase in size to make them easily visible. Similarly, hovering over the document names will also introduce a tooltip above the corresponding circle nodes and increase their size as well.

Mouse click elements: When a user clicks on a document node, it will show a graphic that shows the strength of each topic contributing to the categorization of that document. That is, if a document is projected onto cluster topic *a* and *b*, you will be able to see to which topic the document is more closely related. In cases where only one topic is extracted from the document, the graphic will be entirely a single colour. Refer to Figure 3 for a visual representation of this element.

Omission of certain elements: Rather than overwhelm the user with data and visuals, we chose to reduce the amount of visual information in some regards to simply the visualization. For example, we hide all links (edges) between nodes, instead choosing to use a tooltip to highlight identical document nodes when necessary. While the links (both inner and outer) are hidden, however, they are still used in the calculations for the force-directed graph algorithm. As of now, however, we haven't found a need or a use for the edges so we choose to omit them. Additionally, we choose to hide the topic node, so instead of displaying a node for it at all we simply choose to only show the topic name.

3.5 Project Updates

There are a number of additions that have been made to the previous iteration of this project.

Code Refactoring: I refactored the code on the previous project so that it now runs on Python 3, rather than Python 2, in an attempt to increase the longevity of the work. Additionally, I organized and split up the code where appropriate in order to make the design and structure of DocuVis more modular and maintainable.

Increased Document Support: Whereas VisCluster could only perform the LDA topic modeling with text files, the newer instance of DocuVis has been updated so that it can carry out the topic modeling process with PDFs as well. This raised some issues because many PDFs tend to be encrypted, even the ones that don't need a password to open, but usually the password is just an empty string which some PDF viewers handle on their own. Thus, for encrypted PDFs that prevent the pdfminer.six Python library from extracting their text, we have included a function that will unencrypt these PDFs and prepare them for text extraction.

Streamlined Navigation: One issue that was present in the previous iteration of DocuVis (VisCluster) concerned the task of navigation. As the numbers of documents processed increased, it became increasingly difficult to find a particular document just by looking at the nodes. In the worst case, a user might have to hover over every single node in order to find the document they are looking for. This presents a significant hindrance to any potential user. Thus, in order to correct this issue, we introduced a side navigation panel that allows users to simply search for a document by name. Figure 4 and vis2 show the presence of this navigation panel. Figure 4 demonstrates the mouse hover elements as well, showing that when you hover over a document name, its corresponding node in the graph is enlarged and highlighted.

Side Navigation Panel: The primary purpose of the side panel is to aid in navigation. Upon hovering above any of the document names displayed in the panel, their corresponding nodes in the graph visualization will be enlarged

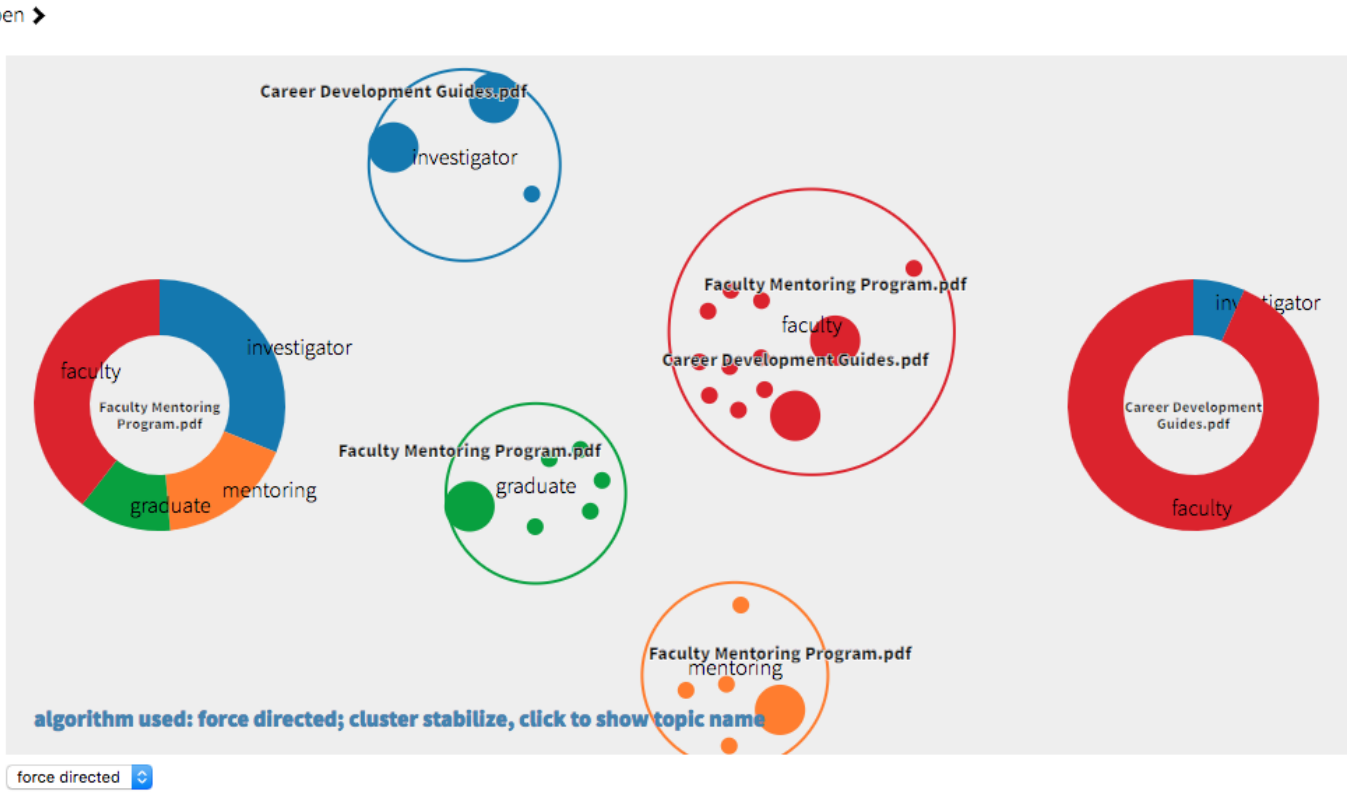

Fig. 3. Clicking on a node will bring up these donut graphics demonstrating the strength of each topic within a certain document.

for ease of identification. However, the panel has a dual purpose of providing extra information about the documents as well. Upon clicking any document names, a dropdown will be displayed containing a list of keywords extracted from the document. An automatically generated summary will also be concurrently displayed under the graph layout display, to help give the user a more detailed understanding of the document from a couple of pertinent sentences. Both the keywords and the summary are generated by the TextRank algorithm described previously. Furthermore, the flexibility of the TextRank algorithm makes it so that the summaries generated can be whatever length we wish to set it to.

These two measures (keywords and summaries) aid in the process of document organization, giving users a sense of what keywords were contained in the document, as well as a quick summary to refresh users' memory of a document if a lot of time has elapsed since they last read it.

Graph Drag: Before the topic cluster circles are drawn to encapsulate the document nodes, it is possible to click and drag around the nodes in the graph layout. This will help to rearrange the position of the nodes if they are not to the user's liking.

4 Short User Survey

Furthermore, we conducted a preliminary user survey to analyze people's ability to generate topics from a set of documents. The purpose of this short survey was to compare human's ability to perform the same task that the LDA topic modeling algorithm is used for, in order to evaluate the performance and usability of the DocuVis visualization system. Given the following document titles:

1) Academic Guidance
2) Advising and Mentoring
3) Career Development Guides
4) Career Resources
5) Diversity, Inclusion, and Equity
6) Faculty Mentoring Program Resources
7) Funding Opportunities for New and Young Faculty
8) Graduate Women's Gatherings
9) Making the Right Moves
10) Helping Students: A Peer Mentoring Approach

We asked five college-educated users between the ages of 21 to 25 to come up with any categories that might be used to accurately and informatively describe the entire set of documents. In other words, we asked them to take on the role of the LDA topic model, but *without access to the documents*. We simply provided them with the document titles, seeing how successful people were in inferring information about a document from the titles. Out of five initial responders, three of them categorized all of these documents into two categories. The two categories these three people suggested were (with one user's generated categories being listed per line) :

- Professional Development, Mentoring
- University, Helping Students
- Academic, Career

A fourth person suggested three categories, those being: Professional Development, Mentoring, and Helping Stu-

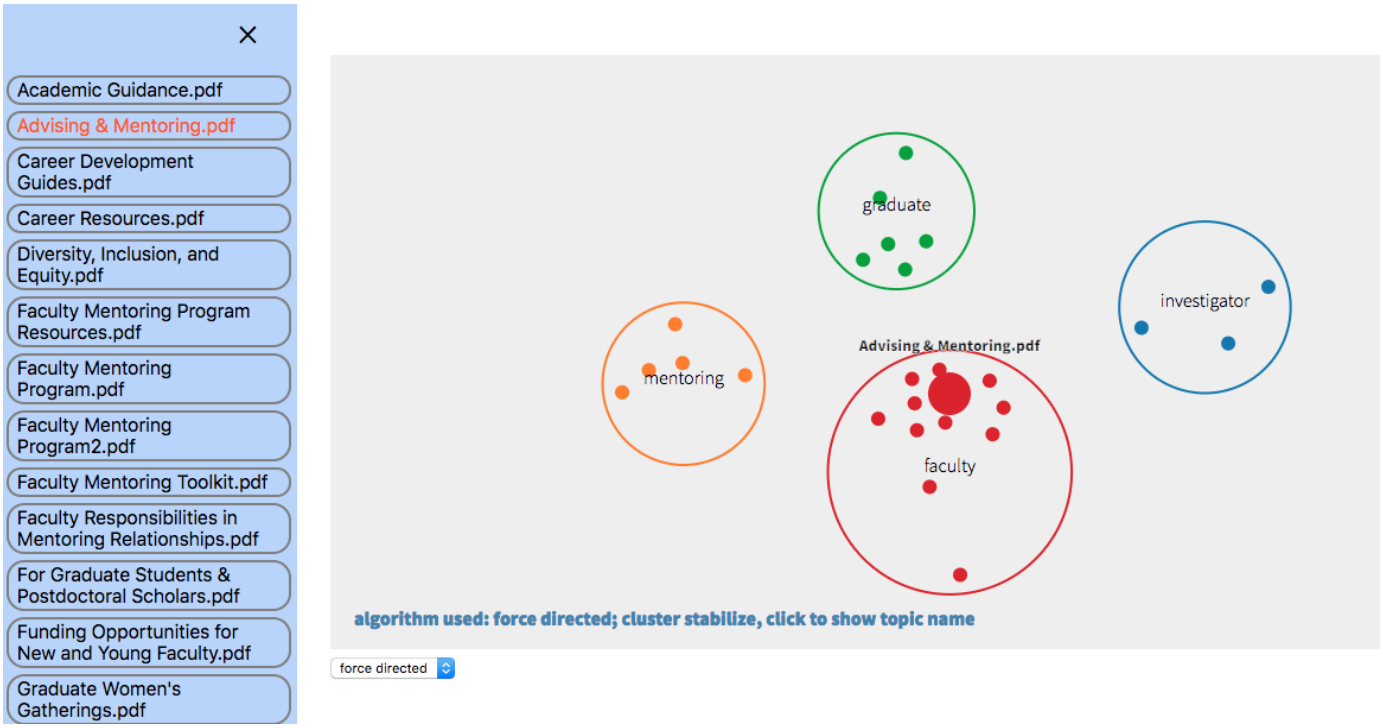

Fig. 4. The document clustering view with the side navigation panel open.

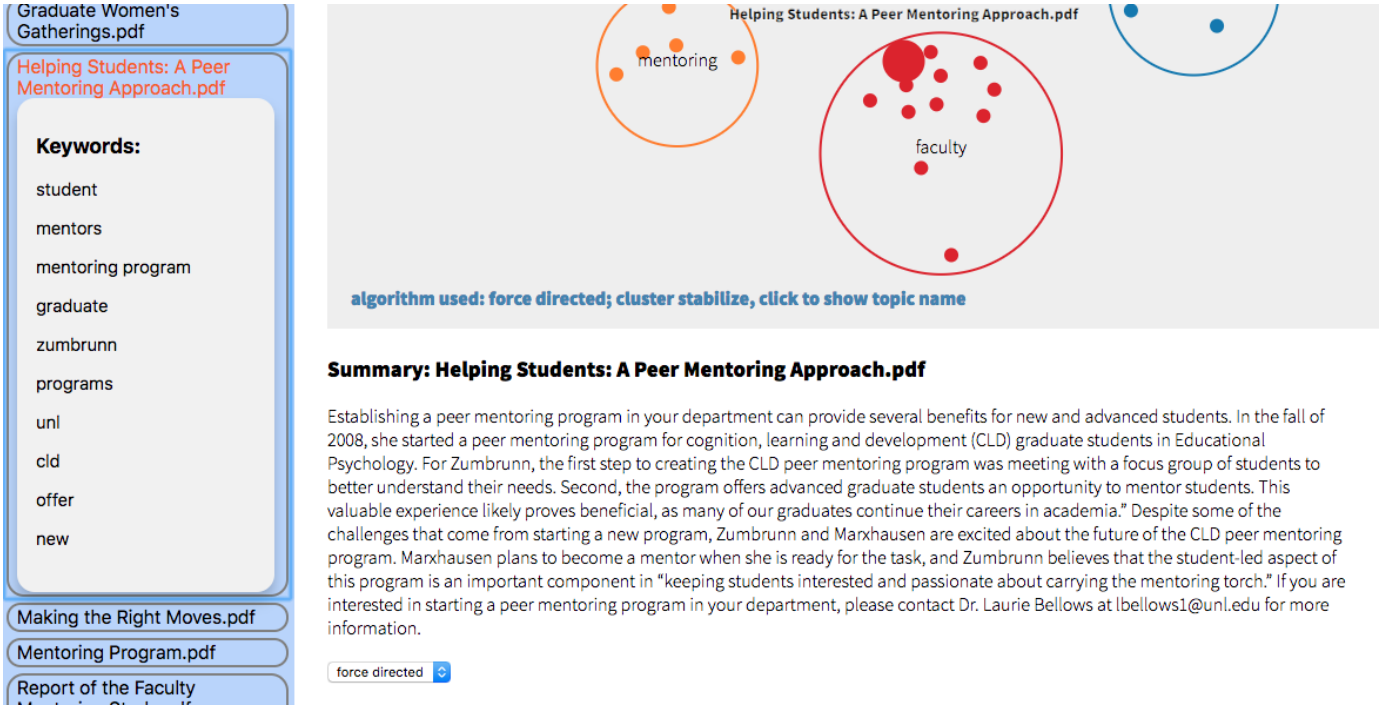

Fig. 5. The side navigation panel shows the dropdown below a document name containing keywords, as well as the corresponding automatically extracted text summary for that document.

dents. Only one person suggested four topics, those being: Career, Faculty, Student, Support.

Naturally, we observe that most of the topic suggestions were words that were included in the titles, with the only exceptions being "University", and "Support". Every other topic suggestion had at least one word show up in the document titles. Similarly, but more importantly, our LDA topic model produced four categories: mentoring, graduate, faculty, and investigator. While three of these topics are still included in the document titles, investigator also highlights a different aspect of the document set that wasn't captured by people because it wasn't clear from the titles. Furthermore, our LDA topic model was able to produce these categories from an increased number of documents: something that becomes more difficult, tedious, and time-intensive for humans to do properly as the number of documents rises.

However, this survey revealed an additional situation where an application of this system may be additionally useful. For example, when users were provided with the name of five papers that fall within the realm of Human-Computer Interaction (HCI) and Crowdsourcing, they were unable to form informative or adequate topic groupings. The paper titles they were given consist of the following:

1) Causeway: Scaling Situated Learning with Micro-Role Hierarchies
2) Participation and Publics: Supporting Community Engagement
3) Supporting Reflective Public Thought with ConsiderIt
4) The Elements of Fashion Style
5) The Anatomy of a Large-Scale Social Search Engine

While they may not seem immediately related, to someone that is familiar and knowledgeable of the literature in HCI and crowdsourcing domains, the connection within this document set should be apparent. However, to users without this knowledge (the five users surveyed had formal education in the humanities, but were unfamiliar of its intersections with more technologically-oriented areas), it was harder to draw a connection between these papers. As such, the categories that people came up with are as follows (again, all the topics created by one person are listed on the same line):

- Learning, Public, Fashion
- Learning Methods, Public Considerations, Fashion
- Fashion, Organizational
- Fashion, Groupings, Technology
- Social Issues, Fashion

In fact, we notice that nobody was able to reconcile *The Elements of Fashion Style* within the other sets of papers. The paper title simply did not provide enough information for people to be able to determine what the document was about, or what kind of information it contained. And in fact, we note that while *The Elements of Fashion Style* does concern fashion related topics, its true research contribution is in producing a polylingual topic model that happens to allow users to express their needs (for an outfit) in natural language in order to receive suggestions from the topic model for appropriate outfits, much in the way a stylist would. Thus, while fashion may be an appropriate topic for this paper, within the current context we can deduce that a grouping under a different topic would have been more appropriate and informative for that document. Thus, we establish an issue that is common among paper and document names: often, without a deep understanding of the content of a document, a title will simply not provide enough information for people to gain an understanding of the document even at a shallow level of categorization.

We also presented the same task with yet another set of documents. This time, the document titles were more lengthy and wordy. They were:

1) Beyond Pro-Choice Versus Pro-Life: Women of Color and Reproductive Justice
2) Patriarchy, the System: An It, Not a He, a Them, or an Us
3) Latinas and the War on Drugs in the United States, Latin America, and Europe
4) White Privilege: Unpacking the Invisible Backpack
5) Queering Antiprison Work: African American Lesbians in the Juvenile Justice System
6) "Night to his Day": The Social Construction of Gender
7) The unforgivable transgression of being Caster Semenya

In this case, the categories generated by people were almost identical:

- Gender, Privilege, Justice, Resistance
- Women's Justice, Work, Gender Ideologies, Racial Ideologies
- Gender, Race
- Race, Gender
- Racial Injustice, Gender/Sexuality Issues

We see that in this particular batch, the ideas that were inferred from the documents were resistance, race/racial injustice, and sexuality issues. Every other topic idea can be directly attributed to the titles of the presented works. It should be noted, however, that some of the responders for this task (the last categorization task in the series of three document sets) forgot to include some of the documents in their categorizations. This gives us reason to believe that making these topic segmentations is not a trivial task, and that the more a person has to do this task, the more prone to error they become. Therefore, it is ideal for a system to do carry out the topic selection process instead, freeing up users cognitive abilities for higher-order organizational tasks.

5 Conclusion

In this paper we presented DocuVis, an interactive visualization system for document clustering and organizing. By incorporating LDA topic modeling and the D3 visualization library, we were able to visualize a force-directed graph to reveal the relationships between a set of documents and extracted topic groupings. Compared to previous work in the field, DocuVis succeeds in highlight inter-document relations, incorporating mechanisms like the multi-level force-directed graph, phrase detection, automatic topic number

selection, navigation tools, and additional document information to provide as much information as possible about the documents while maintaining a clean, aesthetically pleasing visualization.

One of the drawbacks in the existing project, though, is that sometimes the topic names that were generated were not exactly helpful. Looking at Figure 6, for example, we see that the topics generated are: "student", "race", "white", and "people". In this scenario, the documents provided were all related in some way to the Civil Rights Movement of the 1960s. However, we note that the words selected for the topic groupings are not particularly helpful. In particular, "people" is too broad to convey any significance or importance of the material within. The word "white" also shares similar faults in this regard too. Thus one of the drawbacks of this topic modeling approach is that sometimes the words selected as topics are not very helpful at all.

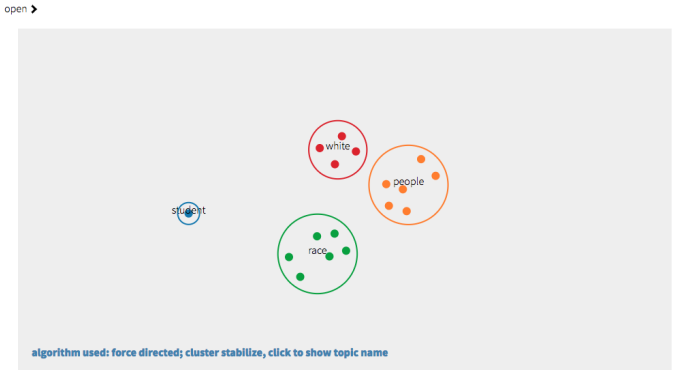

Fig. 6. An example of a suboptimal result from the topic modeling procedure.

In fact, an attempt to rectify this error can be made by including words that we do not wish to be topics as "stop words" that will not be included for consideration as topics. By including the words "white" and "people" as stop words, we are actually able to generate new results, seen in Figure 7.

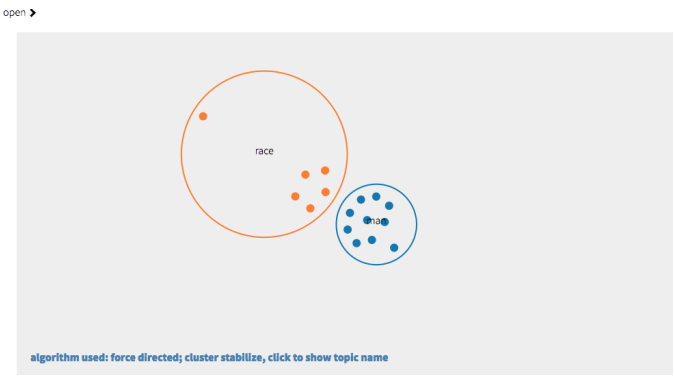

Fig. 7. The resulting topic groupings after the removal of "white" and "people" from topic word considerations.

The danger in doing this, however, is that it may negatively affect the retrieval and grouping of specific topics. And in our scenario, we see that removing people has only served to help the word "man" surface as a topic, which in this context is a similar word and remains as equally uninformative as "people." However, it is possible that through this iterative process, people can begin to exclude words that they do not consider worthy of consideration as topics, and arrive at new topic groupings that the model might not have otherwise produced. This iterative approach is something we leave for future work, but it remains a good idea that before running the LDA topic model, a user can choose any number of words to exclude from topic word consideration so that the model may succeed in generating novel categorizations of document sets. We now move on to talk more about future directions of research to enhance the DocuVis platform.

6 Future Work

There are numerous improvements and directions for future work that can be highlighted. One issue is the issue of working with PDFs. Currently, DocuVis is unable to handle scanned PDFs. Thus, if you take a book and scan a chapter of the book, it will be unrecognizable by DocuVis. This task is more of an optical character recognition problem, but incorporating such a solution into DocuVis might expand the reach of documents we can work with. Furthermore, DocuVis can be a little slower depending on the number of documents provided and the sizes of the documents in question. The preprocessing time for the documents can be quite high by modern standards, requiring anywhere from fifteen to twenty-five minutes depending on the documents provided.

Furthermore, establishing a stronger relationship between the keywords, the summary, and the documents can be important to ease the access of information. For example, linking keywords to their occurring instances in the text can be an easy way to relate the two pieces of information.

Another area of improvement can be had by deploying this application to the web. Currently, the preprocessing is all done offline, and then the results can be observed on a webpage. It might be more beneficial if a user could simply dump their documents into a webpage, have the process be carried out online, and then visualize their results immediately after, without the need to actually run any code. In other words, an enhanced GUI would make this tool more widely accessible for people.

Giving users the option to generate summaries of any given length would also be helpful. Currently, the default length for the automatically generated summaries are 20% of the original document size. This ratio poses a problem as documents increase in size, as a summary that is 20% the length of a 10 page document would still be a two-page summary, which is still too long for a user to quickly scan. Thus, allowing the user to choose custom summary lengths for their document set provides another layer of customization that can support users in their organizational goals. The summaries were also not the best producible summaries in certain scenarios. A generated summary paragraph might have sentences from vastly different sections of the document, resulting in summaries that jumped around and were not exactly coherent. Using a more sophisticated approach to summary generation would be helpful in providing more accurate information about documents. These

improvements may help to make DocuVis even more robust and informative than it already is.

REFERENCES

[1] Charu C. Aggarwal and ChengXiang Zhai. A Survey of Text Clustering Algorithms. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 77–128. Springer US, Boston, MA, 2012.

[2] Federico Barrios, Federico Lpez, Luis Argerich, and Rosa Wachenchauzer. Variations of the Similarity Function of TextRank for Automated Summarization. February 2016.

[3] Dipanjan Das and Andre F T Martins. A Survey on Automatic Text Summarization. page 31.

[4] Ashwinkumar Ganesan, Kiante Brantley, Shimei Pan, and Jian Chen. Ldaexplore: Visualizing topic models generated using latent dirichlet allocation. *arXiv preprint arXiv:1507.06593*, 2015.

[5] Mark Girolami and Ata Kabn. On an Equivalence Between PLSI and LDA. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 433–434, New York, NY, USA, 2003. ACM.

[6] Tomoharu Iwata, Takeshi Yamada, and Naonori Ueda. Probabilistic Latent Semantic Visualization: Topic Model for Visualizing Documents. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 363–371, New York, NY, USA, 2008. ACM.

[7] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, June 2010.

[8] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. iVisClustering: An Interactive Visual Document Clustering via Topic Modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, June 2012.

[9] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, December 2014.

[10] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Texts. page 8.

[11] Jeremy R Millar. Document Clustering and Visualization with Latent Dirichlet Allocation and Self-Organizing Maps. page 6.

[12] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM, 2015.

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

[14] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). In *New results and new trends in computer science*, pages 359–370. Springer, 1991.