HyperMap: a visual analytics approach to analyzing large collections of unstructured text

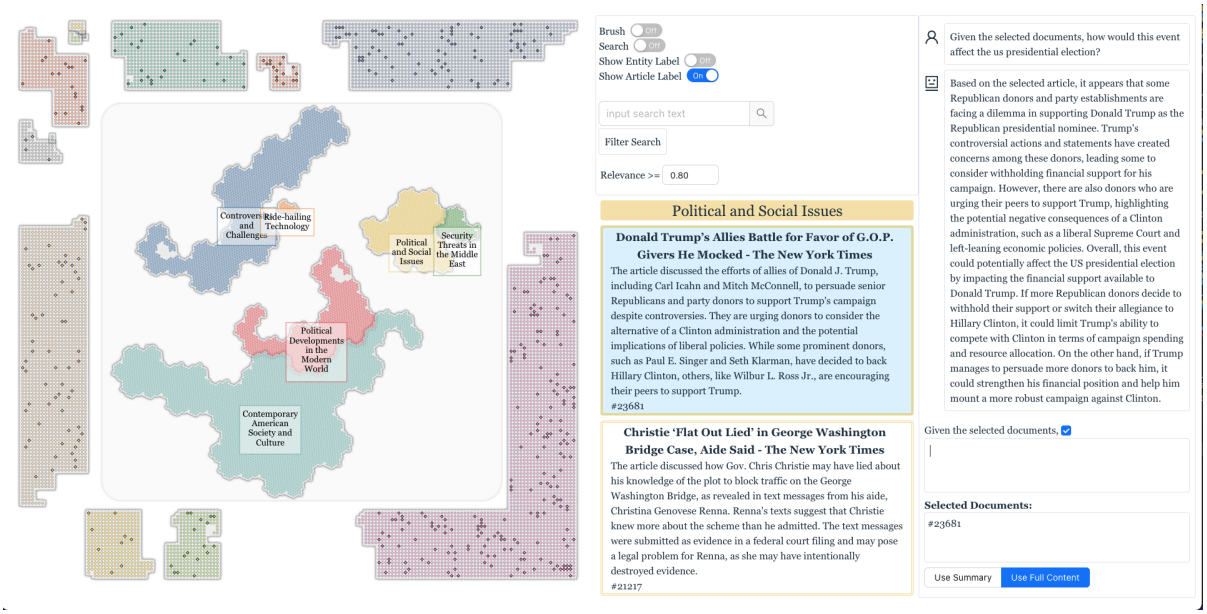


Figure 1: The hypergraph visualization generated by combining generalized Hilbert curves and Gosper curves.

Abstract

Sensemaking on large collections of unstructured text (corpus) is a challenging task that analysts often have to perform. Previous works approach this problem either from a topic- or entity-based perspective, but they lack interpretability and trust. In this paper, we propose HyperMap, a visual analytics approach that combines topic- and entity-based techniques seamlessly by modeling the corpus as a hypergraph. The hypergraph is then hierarchically clustered with an agglomerative clustering algorithm by combining semantic and connectivity similarity. Analysts visualize the clustering result to explore and reorganize a corpus for their analysis. The system is designed to foster interpretability and trust by providing a rich semantic context for the visualization using Large Language Models (LLMs). Case studies and a task-based evaluation are conducted to demonstrate the effectiveness and trustworthiness of the HyperMap system.

Index Terms: Human-centered computing—Visualization—Visualization application domains—Visual Analytics; Information systems—Information retrieval—Retrieval tasks and goals—Clustering and classification; Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Graphical user interfaces; Applied computing—Document management and text processing

1 Introduction

Text data is ubiquitous. From news articles and social media posts to scientific publications, the tremendous amount of text data that is produced poses not only opportunities but also a great challenge to anyone who needs to analyze them. Visual analytics (VA) mitigates this challenge by combining mathematical models and visualizations to automate the sensemaking process and reduce the cognitive load. Chuang et al. [12] proposed that *Model alignment*, the alignment of analysis tasks, visual encodings and model decisions, greatly affects users' interpretation and trust in visual analytic systems. However, in text analysis, the available models often align poorly with analysis tasks. For example, topic models are commonly used to model the topical structure of text documents. Most topic models characterize *topic* as a probabilistic distribution spanning a given vocabulary [42]. This transformation from *topics*, a high-level concept that the user seeks to understand, to a *probabilistic distribution*, a low-level concept that mathematical models can operate on, prevents proper model alignment. The misalignment between analysis tasks and models limits the usage of visual analytics systems for users who are not familiar with the underlying models.

Recent advances in large language models (LLM) present a promising solution to this problem. LLMs have proven successful in various natural language processing (NLP) tasks, especially in question-answering tasks due to their strong capability to understand user intent. Researchers in visualization have adopted LLMs to assist data transformation [44] or directly generate visualization [28]. However, they all assumed a clean data format, where the data to be visualized is already in a table format. For unstructured text analysis though, this is rarely the case. Topics [4], sentiments [7], concepts and entities [9, 34] are common analysis targets in text analysis, which require a data preparation stage to extract them from unstructured text. Recently, Li et al. [26] evaluated ChatGPT's capabilities on Information Extraction (IE) tasks comprehensively, and found that it excels under an OpenIE setting, where the model relies solely on user input to extract information from documents. The capability

of LLMs to extract information from documents according to user intent eliminates the need to carefully align the analysis tasks and models in VA systems, because a specific model is no longer needed to prepare the data for the analysis task. In the previous example, instead of relying on abstruse and unfathomable probabilistic models, LLMs can directly process the text data and summarize the topics of the documents. A user can ask a LLM: *'What are the topics of these articles?'*, and the LLM would give a human-like response, such as *'The articles are about . . . '*.

However, using LLMs in the data preparation stage is not trivial. Problems like *hallucination* and *faithfulness* hinder the accuracy of the extracted information. Token limits restrict the length of the input text, limiting the usage of LLMs on large collections of documents (corpus). Prompts need to be carefully designed to reflect user intent. Finally, the extracted information, the analysis task and the visualization need to be aligned to foster interpretation and trust. In this work, we designed a VA system that models a corpus as a hypergraph, where the nodes are documents and characters (salient entities) extracted from the document using LLMs. We showcase how LLMs are used flexibly to align the data, analysis task and visualization during our design process. The hypergraph is then hierarchically clustered and visualized by extending space-filling curve layouts [29]. The system supports interactive exploration, reorganization and analysis of the documents. To the best of our knowledge, no visual analytics system has adopted LLMs to assist the data preparation stage in text analysis. Using the system, we demonstrate how proper model alignment can be achieved using LLMs.

The contributions of our work are as follows:

- We introduce an LLM-based information extraction pipeline that is capable of extracting topics and salient entities from a given corpus in a way that fosters interpretation.
- We extend space-filling curve layouts to visualize clusters in large hypergraphs.
- We develop a novel VA system that allows users to effectively explore, reorganize and analyze a corpus.

2 Related Works

2.1 Data preparation for large collections of text

Topic-based approaches Topic-based approaches employ certain variations of topic models to organize the documents in a meaningful way. Each topic is often presented as a 'bag of words', which can be in the form of a sequence of words [3, 10, 16, 19, 24, 46, 47] or word clouds [10, 32]. The modeling result provides an overview of the dataset for subsequent analysis tasks. Despite their variations in model choices, the use of topic models as an overview, as well as its 'bag-of-word' visualization, is reported by Lee et al. [25] to be problematic in a comprehensive user study, especially for non-expert users. Chuang et al. [12] concluded that these problems arise from a misalignment between the analysis task, visual encoding and model. The sensemaking process becomes challenging without a basic understanding of the model because the 'bag-of-words' representation is too far away from the user's mental picture of a topic. This misalignment limits the usage of topic models for non-expert users and makes the system prone to produce false positives.

Entity-based approaches A line of work that makes successful model alignments is the entity-based approach. 'Entities' usually include named entities (people, organizations, locations), or meaningful concepts known to an existing knowledge base. The earliest of such approaches is Jigsaw [39], where entities are linked if they appear in the same document. FacetAtlas [9] generalizes the idea of entity to 'facets' which can be entities or any keywords or user's interest. ConceptVector [34] uses 'concept' to represent a similar idea. Generally, entity-based approaches exhibit better model alignments than topic-based approaches [12], but the polysemy of natural language makes them prone to produce false positives [34]. In our work, we use *Characters* to represent entities or concepts that are discussed in the documents. Characters distinguish themselves from previous works in that they are not only mentioned in a document, but they must also be significantly discussed.

Embedding-based approaches Finally, an important line of work organizes documents by directly modeling their semantic similarity [40]. Documents are first projected into a high-dimensional vector space where similarity can be measured, and then a dimensionality reduction technique (e.g.t-SNE) is used to project the dataset onto a two-dimensional space for visualization. Earlier works construct a sparse vector using term-frequency based scores such as *TF-IDF* or BM25 [11, 38]. More recently, the success of pre-trained language models like BERT [15] popularizes the idea of embedding documents in a dense vector space [30, 36, 41]. The embedding can then be used for document retrieval [21, 22] or visualization. Embedding-based approaches also exhibit healthy model alignment, as the vector space directly models the analysis task (finding similar documents). However, the result often lacks explainability and prevents users from trusting the result. Recently, Raval et al. [37] proposed to use LLMs to provide explainability to embeddings-mappings visualizations. We adopt a similar approach in our system to provide explainability to the clustering result.

2.2 LLMs for Information Extraction

Information Extraction aims to identify structured information of interest from unstructured text data. Some of its subtasks include Named Entity Recognition (NER), Relation Extraction (RE) and Event Extraction (EE) [31,45]. Although LLMs have proven successful in many NLP tasks, their application to IE is non-trivial. First, the *faithfulness* of LLMs needs to be carefully evaluated. Faithfulness refers to the ability of a model to adhere to the provided information and not use parametric knowledge learned during training to answer user questions [49]. When conducting information extraction, it is necessary to ensure that the extracted information is actually from the provided text and not from the model's parametric knowledge. Second, the *hallucination* problem of LLMs, where LLMs provide answers factually contradicting to input text (intrinsic) or even factually false (extrinsic). In the context of IE, we mainly focus on the intrinsic hallucination problem. A recent evaluation conducted by Bang et al. [6] found that ChatGPT rarely exhibits intrinsic hallucinations, including the abstractive summarization task from which neural models usually suffer.

More specifically, Li et al. [26] comprehensively evaluated the capabilities of ChatGPT for common IE tasks. They found that ChatGPT excels under the Open-IE setting, where the model relies solely on user input to extract information from documents, but performs poorly under the Standard-IE setting, where ChatGPT is instructed to choose a correct label. Their findings agree with Zhang et al. [48] where ChatGPT is reported to perform poorly on extractive summarization. A common reason for the poor performance of ChatGPT in these tasks is that they are essentially supervised learning tasks, and ChatGPT is not trained to perform them. To make the best use of ChatGPT (or more generally, LLMs) for IE tasks, we need to carefully design the extraction tasks as question-answering tasks instead of supervised learning tasks.

3 Design Rationale

HyperMap is designed for analysts to explore and reorganize a corpus for their analysis. Our design rationale to foster interpretation is based on the design guidelines proposed by Chuang et al. [12]. We

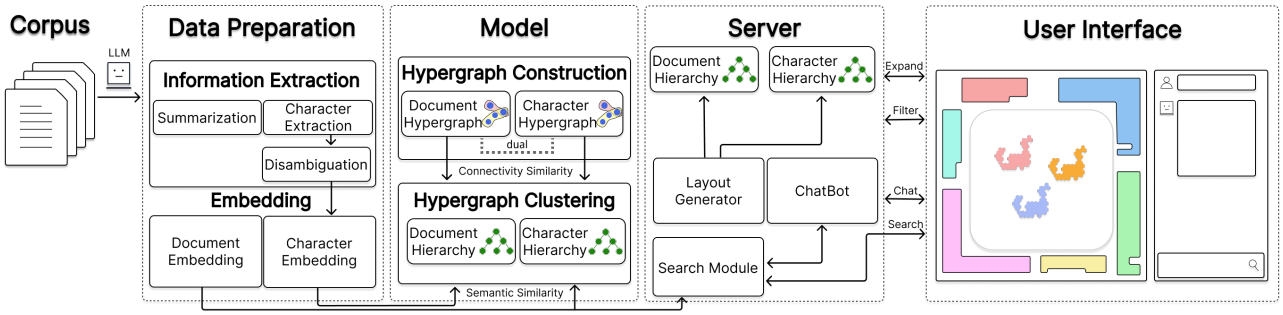


Figure 2: Data processing pipeline of HyperMap. Starting from a corpus of unstructured texts, each document goes through the data preparation stage to extract the main characters. Then the documents and characters are both embedded into a vector space. The model stage constructs a document hypergraph and a character hypergraph, which are then clustered separately by combining connectivity similarity and semantic similarity. The clustered hypergraphs are hosted on the server and visualized in the user interface. Users can expand, filter, or search the hypergraphs to explore the corpus and select documents to be analyzed with a chatbot.

reuse their definitions of *Model Alignment*, *Progressive Disclosure*, and *Unit of Analysis* when describing our design rationale. We first identify common analysis tasks from previous works. Then, we derive our design considerations (DC) from the analysis tasks. We take the DCs into account when making our model decisions and visualization design in Sect. 4. Finally, we explain how we achieve model alignment by applying the design considerations to our system.

3.1 Analysis Tasks

We derive our target analysis tasks from topic- and entity-based approaches. Topic-based approaches aim to support document understanding by visualizing the topic structure of the documents. Investigation of the topic structure seeks to answer the question: *What topics are discussed in the corpus, and how are they related?* Entity-based approaches support investigative analysis by visualizing entities and their relations. We generalize entities to *Characters*, which are the core entities or concepts discussed in the documents. For example, in a news article, the characters can be named entities that appear in the title or are involved in the news event. In a research article, the characters can be the concepts or models proposed by the described work. Similarly, the investigation of the characters seeks to answer the question: *What characters are discussed in the corpus, and how are they related?*: We aim to support both tasks simultaneously as they are fundamental to subsequent tasks and intertwined in a real-world scenario.

3.2 Design Considerations

To support the aforementioned analysis tasks, we derive the following design considerations (DCs):

- **DC1: Overview of topic structures and character connections** Given a corpus, the topic structures and character connections can be complex and cover a wide range of documents and characters. The overview seeks to cover all the documents and characters by hiding the details. This sets the ground for the user to discover their targets of interest.
- **DC2: Progressive Disclosure** To facilitate investigation, it is important to support users to drill down from a high-level overview to intermediate abstractions. This includes disclosure of a specific topic's sub-structure, the containing documents, and the connections to characters.
- **DC3: Model Alignment** Our choice of model should align well with the analyst's mental model when conducting the analysis tasks. This means our model should directly operate on the units of analysis, which are topics (groups of similar documents) and characters. Then by properly visualizing the abstractions of the model, we are safe to produce a good model alignment.
- **DC4: Detailed analysis of the target of interest** The investigation of topic structures and character connections often leads to a target of interest, which can be a topic or a character. After such investigation, previous works usually only provide the user with a list of documents that are relevant to the target of interest. This is perhaps due to the lack of a unified way to analyze the target of interest under different contexts. The advance of LLMs presents a promising solution to this problem by transforming almost any analysis task into a question-answering task. We thus include this task to fill the gap in previous works.

4 Methodology

Starting from a corpus of unstructured texts, we first use LLMs to extract the main characters from each document. The characters are disambiguated and linked to a knowledge base if available. We also create and store the document embeddings and character embeddings. Then, we construct a document hypergraph and a character hypergraph, which are then clustered separately by combining connectivity similarity and semantic similarity. The clustering result is hosted on the server and visualized in an interactive user interface. Below, we describe each component in detail.

4.1 Preprocessing

The Methodology can work for any unstructured dataset

4.1.1 Datasets

[1], [20], [1]

The scope of this research paper is to visualize diverse and unlabelled data conveying any type of information. Our testing is mainly focused on news articles and research papers to show results for all sorts of data. These news articles and abstracts are not domain specific and can be very abstract.
The following datasets were used:

- All The News - This dataset contains 143,000 articles from 15 American publications. It typically includes metadata such as headlines, publication dates, and the full text of the articles. The dataset covers a wide range of topics, making it valuable for various natural language processing and text analysis tasks.

[1] https://platform.openai.com

- Visualization Publications Dataset - This dataset contains information about IEEE visualization publications from 1990-2022. The purpose of using this dataset was to test our methods on majorly technical and scientific datasets. These research papers cover a wide-variety of research topics with fileds like abstracts, keywords, author names, titles ,number of citations, etc. The abstracts are used for the task of event extraction, where the event is the main research idea of the paper.This dataset is used to explore a new kind of event extraction where the definition of an event and main participants change in technical terms as compared to generic articles.

4.1.2 Summarization

Chatgpt for summarization

4.1.3 Document Embedding

OpenAI's embedding API

4.1.4 Character Extraction

Chatgpt for major character extraction and another model for entity linking

4.1.5 Topic Assignment

Chatgpt to assign topics to each cluster [37]

4.2 Models

4.2.1 Hypergraph

A hypergraph is a generalization of a graph in which an edge can connect any number of nodes [17]. A hyperedge thus represents a multi-way relationship between nodes. In our work, we model two types of hypergraphs: document hypergraph and character hypergraph, where documents and characters are the nodes, respectively. Analyzing the document hypergraph and character hypergraph correspond to topic-based and entity-based analysis, respectively. Modeling the corpus as two hypergraphs allows us to support users to conduct both types of analysis simultaneously under a unified framework (**DC1**).

Following the definition of a hypergraph node, a hyperedge can be used to represent two types of multi-way relationships: (1) A hyperedge between *documents* can be constructed between documents that mention the same character. In this case, the hyperedge represents the co-mention of a character, i.e. a named entity or a concept; (2) A hyperedge between *characters* can be constructed between characters if they are mentioned together in the same document. In this case, the hyperedge represents a co-occurrence relationship between characters. Once the two hypergraphs are constructed, they are hierarchically clustered separately. Clusters in the document hypergraph represent topics that are discussed in the dataset. Clusters in the character hypergraph represent characters (entities or concepts) that frequently co-occurred in an article. For better interpretability of the clustering result, we further assign *tags* to each cluster, which is further explained in Sect. 4.1.5.

Although these two types of hyperedges are constructed differently, we utilize the *dual* of a hypergraph to simplify the construction process. The dual of a hypergraph is simply another hypergraph, where the hyperedges are now nodes and the nodes are now hyperedges, as shown in Fig. 3. Therefore, we first model the documents as nodes and characters as hyperedges to construct the document hypergraph H_A. The character hypergraph H_P can then be easily constructed by taking the dual of H_A. This construction process also allows us to use the same clustering algorithm on both hypergraphs, which is further explained in Sect. 4.2.2.

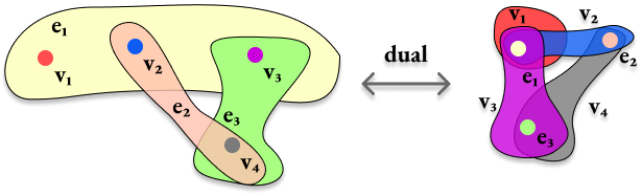


Figure 3: Illustration of the dual of a hypergraph. Left: a hypergraph with 4 nodes v_1, v_2, v_3, v_4 and 3 hyperedges $e_1 = (v_1, v_2, v_3), e_2 = (v_2, v_3), e_3 = (v_3, v_4)$. Right: the dual of the hypergraph, where the nodes and hyperedges interchanged. Nodes in the dual hypergraph are now e_1, e_2, e_3 and hyperedges are $v_1 = (e_1), v_2 = (e_1, e_2), v_3 = (e_1, e_3), v_4 = (e_2, e_3)$.

4.2.2 Hierarchical Clustering

Common clustering algorithms for graphs consider only graph connectivity. However, for the best interpretability of the clustering result, the node embeddings must be also used in the clustering process. The necessity of incorporating node embeddings is further explained in Sect. 4.1.5. Therefore, this limits our choice of clustering algorithms to attributed node clustering algorithms.

Although there are existing approaches that can cluster attributed nodes on graphs such as EVA [13] and iLouvain [14], they are not designed for hypergraphs. In general, hypergraphs can be clustered in two different ways: (1) Directly operate on the hyperedges by generalizing the graph clustering algorithms. For example, Kamiński et al. [8] generalizes the modularity metric for graphs to hypergraphs; (2) First transform the hypergraph into a graph and then apply normal graph clustering algorithms [23]. Although the first approach is more intuitive, it is less scalable and hard to incorporate node attributes. Thus, we have decided to design our clustering algorithm following the second approach.

Considering all the above, we implemented our hierarchical clustering algorithm by first transforming the hypergraph into a graph following the edge re-weighting process proposed by Kumar et al. [23], then an agglomerative clustering algorithm [40] is applied on the re-weighted graph. In agglomerative clustering, the key is to define the similarity between nodes and similarity between clusters. We can easily incorporate node attributes into the clustering process by defining the similarity between nodes and clusters as a combination of attribute similarity S_s and connectivity similarity S_c. Since we're dealing with texts, we refer to the attribute similarity between nodes as semantic similarity.

The semantic similarity $S_s(i, j)$ is the cosine similarity of the embeddings of the two nodes, denoted as v_i. For article nodes, the embeddings are generated using the article content. For participant nodes, the embeddings are generated using a description note of the participant. More details about the embeddings are explained in Sect. 4.1.3. The connectivity similarity S_c is the weighted Topological Overlap (wTO) [18], which is a weighted generalization of the Overlap Coefficient [43], as shown in Equation 1.

$$S_s(i,j) = \frac{v_i \cdot v_j}{||v_i|| \cdot ||v_j||}, \quad S_c(i,j) = \frac{\sum_{u=1}^{N} w_{i,u} w_{u_j} + w_{i,j}}{\min(k_i, k_j) + 1 - |w_{i,j}|} \tag{1}$$

where $k_i = \sum_{j=1}^{N} |w_{i,j}|$ is the total weight of the edges connected to node i. Finally, a weighting factor α is used to balance the two similarities, as shown in Equation 2.

$$S = \alpha S_s + (1 - \alpha) S_c \tag{2}$$

For the similarity between clusters, we used centroid similarity, i.e. the similarity between two clusters is the similarity between the centroids of the two clusters. The algorithm is presented in (TODO: add algorithm pseudocode here)

5 VISUALIZATION

5.1 SFC for HyperGraph

In the design guidelines proposed by Abdelaal et al. [2] in a recent network visualization evaluation study, node-link-based approaches are recommended when: (1) tasks involve the identification of network clusters, and (2) the network is sparse. Condition (1) is fulfilled as explained in Sect. 3, and (2) is guaranteed by the main participant extraction process described in **??**. Therefore, we decided to use node-link-based approaches for our system.

Although there are a variety of node-link-based approaches for hypergraph visualization, we find the extra-node representation proposed by Ouvrard et al. [33] most flexible and intuitive. An extra-node representation improves existing clique-expansion of hypergraphs by adding extra nodes to represent hyperedges. The extra-node representation effectively transforms the hypergraph visualization problem into a bipartite graph visualization problem. After that, any node-link-based graph visualization method can be applied. In our system, we use the space-filling curve (SFC) layout method to layout the extra-node representation of the hypergraph. The SFC layout method uses pre-computed clustering to order nodes in a sequence and then applies a space-filling curve on the node sequence to map it to a two-dimensional screen space [29]. SFC approaches are known for their efficiency and aesthetics in visualizing large graphs [27]. Moreover, SFC layouts support progressive disclosure (**DC2**) organically, as the layout is generated based on the clustering result (**DC3**). After the preprocessing and modeling stage described in Sect. 4, we have two hypergraphs: the article hypergraph H_A and the participant hypergraph H_P, each having its hierarchical cluster. Combining the extra-node representation and SFC layout, we visualize the article hypergraph H_A and the participant hypergraph H_P as two separate SFCs, as shown in Fig. 1.

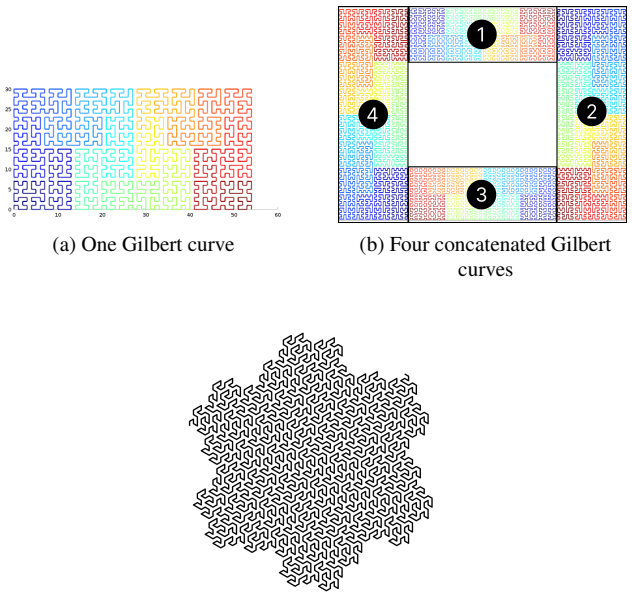
Specifically, we divide the layout space into two parts: the peripheral and the center area. For the peripheral area, we concatenate four generalized Hilbert (Gilbert) curves [50]. A Gilbert curve is a generalized version of the Hilbert curve that can traverse any rectangular region in a way similar to the Hilbert Curve. In Fig. 4a, the Gilbert curve starts from the lower left (dark blue) and ends at the lower right (dark red). Through rotation and flipping, the start and end curve points for neighboring Gilbert curves can be concatenated smoothly, as shown in Fig. 4b. The use of concatenated Gilbert curves allows us to fill the peripheral space while having the efficiency and aesthetics of SFC layouts.

The curve to be used for the center area is technically unbounded. In early prototyping, we found that using the same curve as the peripheral region was confusing for the user, as it was hard to distinguish between the peripheral and center areas. We decided to use a simple Gosper curve (Fig. 4c) to layout the nodes for better aesthetics. The resulting visualization looks similar to GosperMap [5], but we did not employ the advanced techniques proposed in GosperMap. The interactions to support the exploration and reorganization of the dataset are the main focus of the system, which are also not limited to any specific curve.

After the curves are generated, we can apply the curves on the node sequences to generate the two-dimensional layout. We chose to put the article hypergraph in the center area because the articles are the main analysis targets for the user. Consequently, the participant hypergraph is put in the peripheral area.

5.2 Improving the readability

Automatic Cluster Expansion In most cases, the default clustering result is not optimal for the user's targeted analysis tasks. We identify two common problems in early prototyping: (1) clusters may be too big, weakening the semantic meaning that the clusters can convey. (2) clusters may have only one sub-cluster, which makes the parent cluster redundant. Both problems can be mitigated by automatically expanding a cluster. We employ rule-based detection to identify clusters that need to be expanded. For a cluster C, we expand it if the following conditions are met: (1) C has only one sub-cluster C_s; (2) C has more than $n = kN$ nodes, where $k \in [0,1]$ and N is the size of the hypergraph. Through trial and error, we find that $k = 0.3$ gives the most balanced results.



(a) One Gilbert curve

(b) Four concatenated Gilbert curves

(c) An order-4 Gosper curve

Figure 4: (a), (b): Illustration of concatenating Gilbert curves. The color indicates the traverse direction of Gilbert curves: start with dark blue and end with dark red. (c): An example of order-4 Gosper curve used for the center area.

Spacing Strategy Spacing between each node is important for the readability and aesthetics of SFC layouts. To highlight different clusters, we employ our spacing strategy on clusters instead of nodes. Given a space-filling curve of a specific order, we first calculate the length of the curve L. L represents the total amount of space available for the nodes and thus, $L-N$ represents the amount of space to be redistributed, where N is the size of the hypergraph. Our goal is to distribute the space between clusters to ensure the best readability. In early prototyping, we found that distributing the space proportional to the cluster size gives the best readability as well as stability. Specifically, we define the space of a cluster as the blank space it has behind it on the curve, which is calculated by $(L-N)\frac{N_c}{N}$, where N_c is the size of the cluster (Fig. 5). Another consideration when designing our spacing strategy is stability. We want to ensure that the layout change is minimized when clusters are expanded. Naively, when a cluster is expanded, the whole layout needs to be recalculated because now the cluster sequence is changed. To avoid such recalculation, we design our spacing strategy in this way so that the sub-clusters can simply take over the space of their parent cluster, as shown in Fig. 5. Since the total volume of the sub-clusters is the size of the parent cluster and the space is proportional, the sub-clusters can take over exactly the space of the parent cluster without any overflow or underflow. This ensures a local change in the layout when a cluster is expanded.

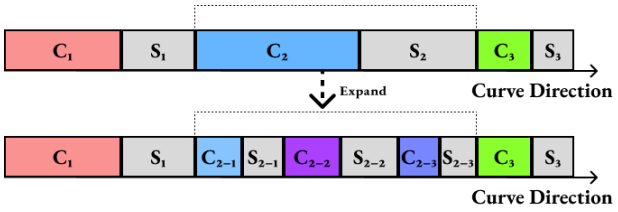
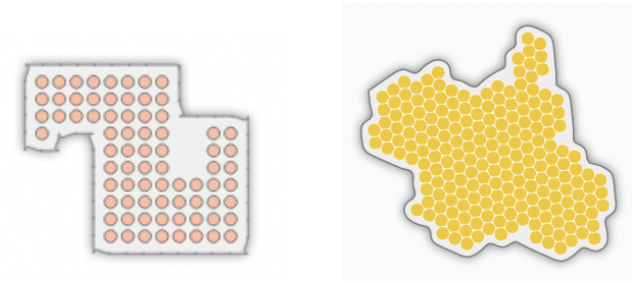


Figure 5: Spacing strategy of the SFC layouts. The space of each cluster S_i is proportional to the cluster size. When C_i is expanded, its sub-clusters redistribute S_i to ensure the expansion only affects locally.



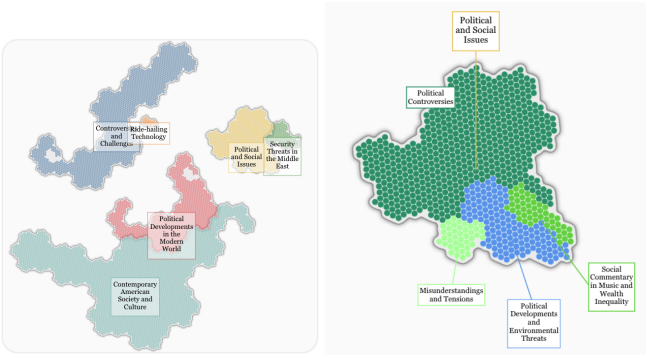
(a) (b)

Figure 6: (a) An example of Gilbert cluster border (b) An example of Gosper cluster borders

Concave hull approximation After applying the SFC layout, we use a concave hull algorithm [35] to generate an approximation polygon for each cluster. The polygons are used to generate borders and calculate label positions for the clusters. The concave hull algorithm is generated based on a cluster of points, in our case, the nodes in a cluster. This means that the algorithm can be applied to any curve we used for the SFC layout. This is a desirable property because we are using two different curves in our system, and the center area curve choice is flexible. Using the same algorithm guarantees a unified aesthetic across curves.

The original concave hull algorithm is designed for approximating points, but we need to approximate circles. Naively we can use the center of the circles as the points for the algorithm, but this would result in a polygon that is too small and crosses the circles on the boundary. To address this issue we use a simple trick: we add extra points to the cluster by extrapolating the original points. For a Gilbert curve, the curve moves perpendicularly, so the resulting polygon would have perpendicular corners. We can therefore add eight extra points around each original point so that the extrapolation forms a three-by-three grid. On the boundary of the cluster, these extra points prevent the concave hull from passing across the original points. Since the concave hull algorithm has an $O(n \log n)$ time complexity, the performance overhead introduced by the extra points is negligible.

Borders The borders are generated by applying a smoothing algorithm on the polygons. For Gosper curves, we use the polygon as control points to generate a cubic basis spline as the border. For Gilbert curves, we use a similar approach but with a cubic Bezier curve. More specifically, for each pair of consecutive points, we use a smoothing factor to interpolate the control point. This results in a sketchy style at the border corners. Two examples are given in Fig. 6.



(a) Label position of the cluster is the centroid of each cluster.

(b) Labels of the sub-clusters of a cluster. The color indicates different sub-clusters.

(c) Labels of the character cluster. The left part shows the categories of the characters as the cluster label. When documents are selected, the right part appears and shows the connected characters.

Figure 7: (a) Labels (topics) of the document clusters (b) An example of expanded cluster labels (c) An example of character labels (categories) and highlighted characters

Labeling Labeling the clusters is essential for users to explore the dataset. We use the topic assignment described in Sect. 4.1.5 to label the clusters. When using SFC layouts, determining the label position automatically is challenging because the shape of the clusters can be irregular. For a cluster, the label position is simply the centroid of the polygon. When a cluster is expanded through user interaction, the sub-clusters within need to be clearly labeled as well. Using the centroid of the sub-cluster as the label position is not a good choice because the label would cause a serious cluttering issue. Therefore, for a sub-cluster, we first calculate the centroid of the sub-cluster, and then we extend the line from the parent cluster centroid to the sub-cluster centroid. Once the intersection point of the extended line and the parent border is found, we extend the line by a fixed amount to avoid any overlapping issues. This results in a radial layout for the sub-cluster labels, as shown in Fig. 7. The generalizability of the concave hull algorithm makes our labeling position calculation applicable to any curve we use for the SFC layout.

6 HyperMap System Design

Below, we describe the HyperMap frontend system, including the views, visualizations and interactions, and how they assist exploration and reorganization of a corpus. The HyperMap system consists of three main views: Cluster View, Article View and Analysis View. The Cluster View visualizes the corpus as two hypergraphs using the SFC layout described in Sect. 5. The Document View shows the articles along with necessary statistics when the user makes a selection in the Cluster View. The Analysis View integrates an LLM-based chatbot to assist the user in analyzing the selected articles. **TODO: add labels to each view in figure 1**

6.1 Cluster View

Cluster View is the main view of the HyperMap system. It visualizes the corpus as two hypergraphs using the SFC layout described in Sect. 5. Using the Cluster View, users can explore the topic structure and entity connections simultaneously ((DC1)). Below, we mainly discuss user interactions and the coordination between the Cluster View and other views.

6.1.1 Interactions

Hover and click By default, the cluster labels are hidden to reduce clutter. Hovering over the clusters will trigger a highlight effect and show the cluster label, indicating to users that it is an interactive object. Users can select a cluster by clicking on the cluster label. This will also trigger the Article View to show articles within the cluster and the connected entities (**DC1**). Additionally, clicking on the cluster will temporarily expand the cluster to expose its sub-structure (**DC2**), as shown in Fig. 7b. The sub-structure is colored in different colors while maintaining the original cluster's shape. The labels of each sub-cluster are also shown radially. When the user clicks on any cluster or cluster label, the connected characters are highlighted and others fade out. Under such cases, hovering over the character clusters will show not only the cluster label but also the highlighted characters in a list, as shown in Fig. 7c. Users can click on the connected characters to filter the articles in the Article View.

Expansion The expansion operation breaks a cluster into smaller clusters. This operation is necessary because the agglomerative clustering result is not always semantically optimal. Some clusters may be too vague, while others may be too specific. By breaking down a cluster, users can also investigate a level deeper into the topic structure (**DC2**). We use (Cmd + Click) or (Ctrl + Click) to expand a cluster. The sub-cluster will redistribute the spacing of their parent cluster proportionally to their size. This ensures that other clusters are not moved in the layout. **TODO: figure**

Filtering At any point in the exploration when users find that they have found the target of interest, they can click the filter button to remove irrelevant documents and characters from the view. The filtering functionality effectively creates a sub-hypergraph based on node selection, which supports any interactions that are available in the original hypergraph. Note that we only support filtering based on document node selection, and character nodes that are not connected to the selected document nodes are filtered accordingly. Although filtering based on character nodes is technically possible, we do not find the operation intuitive. We decided to remove this feature to prevent users from losing themselves in the exploration process.

Searching HyperMap supports searching by document embeddings. Users can create any query in natural language, and the system will return the most relevant documents. The search functionality is implemented by ranking the documents based on their cosine similarity to the query. The user query is first embedded into the same vector space as the documents. Then the cosine similarity between the query and each document is calculated. The server returns the ranked documents to the front end, and the user can control the number of documents to be highlighted by a relevancy threshold. The highlighted documents and connected characters are visually distinguished in the Cluster View. **TODO: figure**

6.2 Document View

The Document View displays a list of articles in a cluster. Users can click any cluster or sub-cluster label to inspect the articles within. The document view is colored according to the cluster color in the Cluster View. Each document is an interactive card, with the title, summary, ID and relevancy score (if available) shown. If the user is under search mode, documents above the relevancy threshold are also highlighted. Users can click any document card to add or remove it in the Analysis View.

6.3 Analysis View

The Analysis View is a chatbot that assists users in analyzing the selected documents (**DC4**). Users can use the summary or the full content of the selected document to ask any questions in natural language. For example, ...

7 Evaluation

7.1 Case Study

7.1.1 AllTheNews Dataset

7.1.2 Vis Publication Dataset

References

[1] All The News. `https://components.one/datasets/all-the-news-2-news-articles-dataset`.

[2] M. Abdelaal, N. D. Schiele, K. Angerbauer, K. Kurzhals, M. Sedlmair, and D. Weiskopf. Comparative evaluation of bipartite, node-link, and matrix-based network representations. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):896–906, 2022.

[3] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, and M. Gleicher. Serendip: Topic model-driven visual exploration of text corpora. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 173–182, 2014. doi: 10.1109/VAST.2014.7042493

[4] D. Atzberger, T. Cech, W. Scheibel, M. Trapp, R. Richter, J. Döllner, and T. Schreck. Large-scale evaluation of topic models and dimensionality reduction methods for 2d text spatialization. *arXiv preprint arXiv:2307.11770*, 2023.

[5] D. Auber, C. Huet, A. Lambert, B. Renoust, A. Sallaberry, and A. Saulnier. Gospermap: Using a gosper curve for laying out hierarchical data. *IEEE transactions on visualization and computer graphics*, 19(11):1820–1832, 2013.

[6] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[7] Z. J. Beasley, A. Friedman, and P. Rosen. Through the looking glass: insights into visualization pedagogy through sentiment analysis of peer review text. *IEEE Computer Graphics and Applications*, 41(6):59–70, 2021.

[8] P. P. . F. T. Bogumił Kamiński. Community detection algorithm using hypergraph modularity. In *Complex Networks & Their Applications IX: Volume 1, Proceedings of the Ninth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2020*, pp. 152–163. Springer, 2021.

[9] N. Cao, J. Sun, Y.-R. Lin, D. Gotz, S. Liu, and H. Qu. Facetatlas: Multifaceted visualization for rich text corpora. *IEEE transactions on visualization and computer graphics*, 16(6):1172–1181, 2010.

[10] I. Cho, W. Dou, D. X. Wang, E. Sauda, and W. Ribarsky. Vairoma: A visual analytics system for making sense of places, times, and events in roman history. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):210–219, 2016. doi: 10.1109/TVCG.2015.2467971

[11] J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013. doi: 10.1109/TVCG.2013.212

[12] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 443–452, 2012.

[13] S. Citraro and G. Rossetti. Eva: Attribute-aware network segmentation. In *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8*, pp. 141–151. Springer, 2020.

[14] D. Combe, C. Largeron, M. Géry, and E. Egyed-Zsigmond. I-louvain: An attributed graph clustering method. In *Advances in Intelligent Data*

Analysis XIV: 14th International Symposium, IDA 2015, Saint Etienne. France, October 22-24, 2015. Proceedings 14, pp. 181–192. Springer, 2015.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] W. Dou, L. Yu, X. Wang, Z. Ma, and W. Ribarsky. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2002–2011, 2013.

[17] M. T. Fischer, A. Frings, D. A. Keim, and D. Seebacher. Towards a survey on static and dynamic hypergraph visualizations. In *2021 IEEE visualization conference (VIS)*, pp. 81–85. IEEE, 2021.

[18] D. M. Gysi, A. Voigt, T. d. M. Fragoso, E. Almaas, and K. Nowick. wto: an r package for computing weighted topological overlap and a consensus network with integrated visualization tool. *BMC bioinformatics*, 19(1):1–16, 2018.

[19] D. Han, G. Parsad, H. Kim, J. Shim, O.-S. Kwon, K. A. Son, J. Lee, I. Cho, and S. Ko. Hisva: A visual analytics system for studying history. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4344–4359, 2022. doi: 10.1109/TVCG.2021.3086414

[20] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko. vispubdata.org: A metadata collection about IEEE visualization (VIS) publications. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2199–2206, Sept. 2017. doi: 10.1109/TVCG.2016.2615308

[21] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

[22] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781. Association for Computational Linguistics, Online, Nov. 2020. doi: 10.18653/v1/2020.emnlp-main.550

[23] T. Kumar, S. Vaidyanathan, H. Ananthapadmanabhan, S. Parthasarathy, and B. Ravindran. A new measure of modularity in hypergraphs: Theoretical insights and implications for effective clustering. In *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8*, pp. 286–297. Springer, 2020.

[24] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. In *Computer graphics forum*, vol. 31, pp. 1155–1164. Wiley Online Library, 2012.

[25] T. Y. Lee, A. Smith, K. Seppi, N. Elmqvist, J. Boyd-Graber, and L. Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies*, 105:28–42, 2017.

[26] B. Li, G. Fang, Y. Yang, Q. Wang, W. Ye, W. Zhao, and S. Zhang. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*, 2023.

[27] K.-L. Ma and C. W. Muelder. Large-scale graph visualization and analytics. *Computer*, 46(7):39–46, 2013. doi: 10.1109/MC.2013.242

[28] P. Maddigan and T. Susnjak. Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models. *IEEE Access*, 2023.

[29] C. Muelder and K.-L. Ma. Rapid graph layout using space filling curves. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1301–1308, 2008.

[30] A. Narechania, A. Karduni, R. Wesslen, and E. Wall. vitality: Promoting serendipitous discovery of academic literature. 2022.

[31] Z. Nasar, S. W. Jaffry, and M. K. Malik. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)*, 54(1):1–39, 2021.

[32] D. Oelke, H. Strobelt, C. Rohrdantz, I. Gurevych, and O. Deussen. Comparative exploration of document collections: a visual analytics approach. In *Computer Graphics Forum*, vol. 33, pp. 201–210. Wiley Online Library, 2014.

[33] X. Ouvrard, J. L. Goff, and S. Marchand-Maillet. Networks of collaborations: Hypergraph modeling and visualisation. *CoRR*, abs/1707.00115, 2017.

[34] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist. Conceptvector: Text visual analytics via interactive lexicon building using word embedding. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):361–370, 2018. doi: 10.1109/TVCG.2017.2744478

[35] J.-S. Park and S.-J. Oh. A new concave hull algorithm and concaveness measure for n-dimensional datasets. *Journal of Information science and engineering*, 28(3):587–600, 2012.

[36] R. Qiu, Y. Tu, Y.-S. Wang, P.-Y. Yen, and H.-W. Shen. Docflow: A visual analytics system for question-based document retrieval and categorization. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[37] S. Raval, C. Wang, F. Viégas, and M. Wattenberg. Explain and trust: An interactive machine learning framework for exploring text embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[38] E. Sherkat, S. Nourashrafeddin, E. E. Milios, and R. Minghim. Interactive document clustering revisited: A visual analytics approach. In *23rd International Conference on Intelligent User Interfaces*, pp. 281–292, 2018.

[39] J. Stasko, C. Gorg, Z. Liu, and K. Singhal. Jigsaw: Supporting investigative analysis through interactive visualization. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 131–138, 2007. doi: 10.1109/VAST.2007.4389006

[40] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. 2000.

[41] Y. Tu, O. Li, J. Wang, H.-W. Shen, P. Powałko, I. Tomescu-Dubrow, K. M. Slomczynski, S. Blanas, and J. C. Jenkins. Sdrquerier: A visual querying framework for cross-national survey data recycling. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[42] I. Vayansky and S. A. Kumar. A review of topic modeling methods. *Information Systems*, 94:101582, 2020.

[43] M. Vijaymeena and K. Kavitha. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 3(2):19–28, 2016.

[44] C. Wang, J. Thompson, and B. Lee. Data formulator: Ai-powered concept-driven visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[45] W. Xiang and B. Wang. A survey of event extraction from text. *IEEE Access*, 7:173111–173137, 2019.

[46] Y. Yan, Y. Tao, S. Jin, J. Xu, and H. Lin. An interactive visual analytics system for incremental classification based on semi-supervised topic modeling. In *2019 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 148–157, 2019. doi: 10.1109/PacificVis.2019.00025

[47] Y. Yang, Q. Yao, and H. Qu. Vistopic: A visual analytics system for making sense of large document collections using hierarchical topic modeling. *Visual Informatics*, 1(1):40–47, 2017.

[48] H. Zhang, X. Liu, and J. Zhang. Extractive summarization via chatgpt for faithful summary generation. *arXiv preprint arXiv:2304.04193*, 2023.

[49] W. Zhou, S. Zhang, H. Poon, and M. Chen. Context-faithful prompting for large language models, 2023.

[50] J. Červený. https://github.com/jakubcerveny/gilbert/commits/master generalized hilbert ("gilbert") space-filling curve for rectangular domains of arbitrary (non-power of two) sizes., 2019.