Event HyperGraph for analyzing large collections of text

Sam Yu-Te Lee*
University of California, Davis

Kwan-Liu Ma†
University of California, Davis

ABSTRACT

Sensemaking on large collections of text is a challenging task that analysts 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 a visual analytics system that allow analysts to explore and reorganize a corpus to suit their needs and quickly make sense of the 4Ws from the organized corpus. The system first organizes a large corpus into a hypergraph by combining topic- and entity-based extraction techniques. Then the hypergraph is hierarchically clustered and visualized for analysts to explore and reorganize interactively. Finally, an event graph visualization method based on storyline enables analysts to quickly make sense of the 4Ws. The whole pipeline is designed to foster interpretability and trust by providing semantic context of the visualization and by supporting curating interactions. Case studies and a task-based evaluation are used to demonstrate the effectiveness and trustworthiness of the system.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Treemaps; Human-centered computing—Visualization—Visualization design and evaluation methods

1 INTRODUCTION

2 RELATED WORKS

2.1 Hypergraph visualization

[1]

2.2 Summarizing large collections of text

Topic models, entity-based summarization (VA approaches)

2.3 Interpretability and Trust in text analysis

3 DESIGN RATIONALE

What DRs are needed for analysts to explore and reorganize a corpus for their analytical tasks and why?

3.1 Design Considerations

- **DC1: An overview of the topic structure**
- **DC2: Support for curation through user interaction**
- **DC3: Detailed analysis of the curated result**

4 METHODOLOGY

4.1 Modeling

4.1.1 Hypergraph Construction

A hypergraph is a generalization of a graph in which an edge can connect more than two nodes [**?**]. A hyperedge thus represents a multi-way relationship between nodes. In this paper, we model two types of hypergraphs: article hypergraph and participant hypergraph, where articles and participants are the nodes, respectively. *Participants* are the core components that the article's content discuss [**?**]. For example, in a news article, the participants can be named entities such as people, organizations, or locations. In a research article, the participants can be the concepts or techniques used in the article.

Following the definition of a hypergraph node, a hyperedge can be used to represent two types of multi-way relationships: (1) A hyperedge between *articles* can be constructed if the articles all mention the same participant. In this case, the hyperedge represents the co-mention of a participant, i.e. a named entity or a concept; (2) A hyperedge between *participants* can be constructed if the participants are mentioned together in the same article. In this case, the hyperedge represents a co-occurrence relationship between participants.

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. (Add formulas here to explain). Therefore, we first model the articles as nodes and participants as hyperedges to construct the article hypergraph H_A. Then we apply a hierarchical clustering algorithm on H_A. The detail of the clustering algorithm is explained in Sect. 4.1.2. The result of the clustering algorithm represents topics that are discussed in the articles. Then, we apply the same clustering algorithm on the dual of the hypergraph, which is the participant hypergraph H_P. The result represents groups of participants that are frequently mentioned together in the articles.

4.1.2 Hierarchical Clustering

We implement our hierarchical clustering algorithm following the agglomerative clustering approach. We combine the semantic similarity S_s and connectivity similarity S_c to calculate the distance between two nodes. $S_s(i,j)$ is calculated using the cosine similarity of the embeddings of the two nodes. For article nodes, the embeddings are simply the document embeddings of the articles. For participant nodes, the embeddings are the average of the embeddings of the articles that mention the participant. S_c is calculated using the Jaccard similarity of the two nodes' neighbors (Equation 1).

$$S_c(i,j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|} \tag{1}$$

A weighting factor α is used to balance the two similarities, as shown in Equation 2.

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

4.1.3 Topic Assignment

Chatgpt to assign topics to each cluster

4.2 Preprocessing

The Methodology can work for any unstructured dataset

4.2.1 Summarization

Chatgpt for summarization

4.2.2 Document Embedding

OpenAI's embedding API

*e-mail: ytlee@ucdavis.edu
†e-mail: klma@ucdavis.edu

4.2.3 Participant Extraction

Chatgpt for major participant extraction and another model for entity linking

5 VISUALIZATION

5.1 Space Filling Curves

Introduce Gosper curve and generalized Hilbert curve, and how they are used for large graph layout

5.2 SFC for HyperGraph

Using the Gosper curve to layout the article graph

Concatenating four generalized Hilbert curve to layout the entity graph on the peripheral

5.3 Spacing Strategy

5.4 Border Approximation

5.5 Edge Bundling

ACKNOWLEDGMENTS

The authors wish to thank A, B, and C. This work was supported in part by a grant from XYZ.

REFERENCES

[1] 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.