Brandeis Capstone Project Report

HAYDEN MCCORMICK

May 2024

1 Abstract

This report presents the MMIF Graph Visualizer, a highly interactive visualization tool for exploring, searching, and analyzing large collections of MultiMedia Interchange Format (MMIF) files. The aim of this project is to allow for immediate high-level insights into a set of documents by rendering a corpus as a set of physics-based force-directed nodes linked by shared entities, enhanced by abstractive summarization, clustering, and topic modeling. Because visualization is a historically difficult field to quantitatively evaluate, I also discuss potential metrics for measuring the usefulness of a visualization.

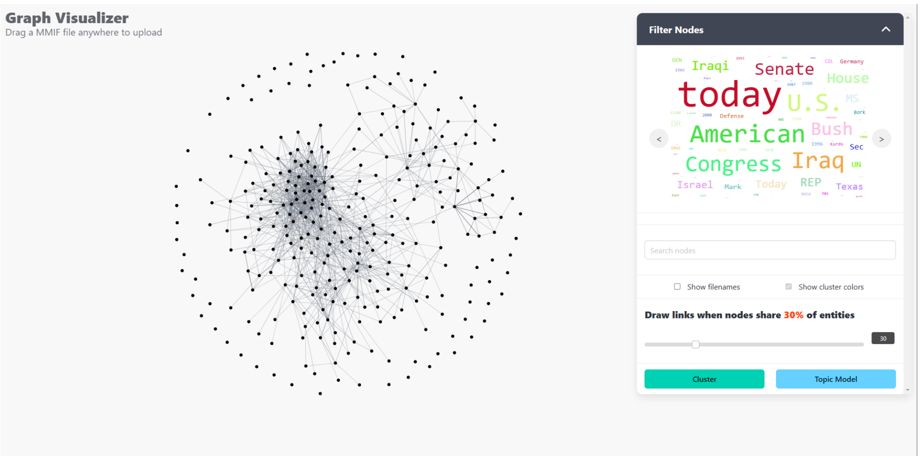
2 Introduction

2.1 Motivation

The MMIF format allows a set of documents, generally corresponding to a single archive video, to be directly annotated by an arbitrarily large set of natural language processing and computer vision applications. Although this is an efficient and convenient way for programs to access this metadata, a common complaint with MMIF is that an archivist needs to skim through tens (or even hundreds) of thousands of JSON lines to understand what a file actually represents. From this frustration came the MMIF visualizer, which aims to faithfully present the *context* of a MMIF file by rendering the documents it refers to, and the *content* by rendering several HTML visualizations corresponding to each annotation type (e.g., thumbnails for OCR annotations).

Although this tool makes a significant difference in a user's ability to understand a single MMIF file, another problem remains. These files are generally run in batches, and filenames often refer to the GUID of the source documents, making exploration and search incredibly difficult at the document-set level.

This motivates the concept of an effective visualizer which operates on the collection-level. Specifically, unlike other approaches, the Graph Visualizer aims to leverage the fact that certain documents can be intuitively recognized as more "related" than others – by sharing themes, event mentions, or people, for example.



Search nodes

□ Show filenames ☑ Show cluster colors

Draw links when nodes share 30% of entities

30

Cluster

Topic Model

Figure 1: Screenshot of Graph Visualizer with 300 nodes

2.2 Objectives

To this end, the project has the following goals:

1. Representing an arbitrarily large set of files spatially, so that relationships and clusters of files can be easily and intuitively identified.

2. Improving ease of search through a body of MMIF files, and to implement search/filtering not just for document-finding, but also for identifying new patterns and connections in a narrowed search space. Focus especially on interactivity.

3. Adding explainability by representing each node – or sets of nodes – as abstractive document summaries.

3 Related Work

Document visualization is a well-established problem in NLP, and has inspired many approaches. The most common and natural visualization is to embed each document in a common embedding space, projecting down to a lower dimension using techniques such as PCA [4]. Further work has been done in representing documents as semantically dense cards [7], which allows for significantly quicker understanding of a source document at a glance. This project also draws on concepts from topic modeling [6] and knowledge graphs [5], both of which are techniques for spacially representing the semantic relationships between bodies of text.

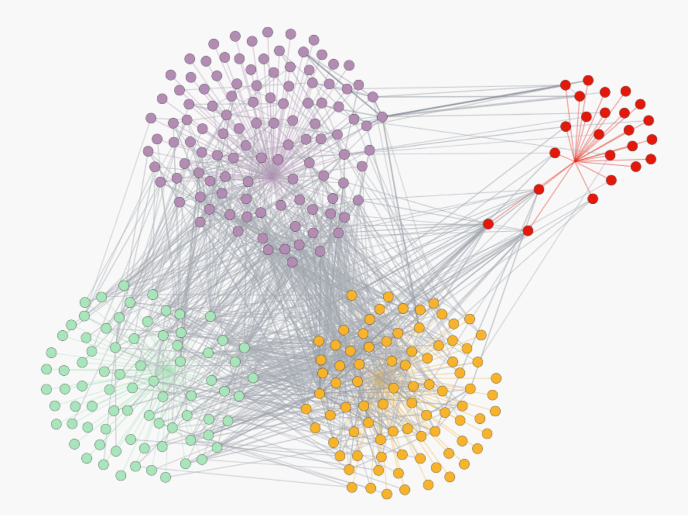


Figure 2: Clustering with entity-linked nodes

4 Visualization

The core of the Graph Visualizer's visualization suite is the node document representation with entity links; every filtering, searching, and clustering operation in some way manipulates one or several of the nodes. Each node, when clicked, contains a short abstractive summary of its document content, and edges are established between nodes based on the number of shared entities in their transcript. The nodes are rendered using D3, a robust Javascript visualization suite.

The result of this is an interactive representation akin to PCA document embedding projections, where natural ”clusters” are formed based on common groups of entities. Because of link-based forces in the D3 simulation, more ”central” nodes which share the most common entities are generally forced to the center, whereas outliers appear on the outside.

The visualizer also contains two cluster views which apply separate models/algorithms: a K-means algorithm and a BERT-based topic model. Since the entity edges remain intact during clustering, nodes which share many entities with neighboring clusters are pulled towards the center, giving a visual analogue to cluster coherence by node.

5 Modeling/Evaluation

This section describes the modeling techniques employed by the Graph Visualizer, and evaluates each using a quantitative metric. Some more trivial models, such as SPACY for named entity recognition, are also included in the program.

5.1 Summarization

Although the latest large language models achieve state-of-the-art performance in summarization[8], they are often slow if run locally, or costly otherwise. On the other hand, non-attention-based approaches can work very well for extractive text summarization, but struggle with (abstractive) generation. A good middle ground is BART, an encoder-decoder transformer inspired by BERT which excels especially at summarization after fine-tuning.

For generating abstractive summaries, I used bart-large-cnn[1], a BART checkpoint fine-tuned on CNN Dailymail news articles. Since this project is largely centered around the domain of AAPB/NewsHour videos, I first wanted to evaluate the model's performance against AAPB document descriptions, the closest thing to ground-truth summaries. The model does not excel at this evaluation, with a 0.091726 ROUGE-1 score, though this is unsurprising given the rigid structure of AAPB descriptions, which each include a short sentence about the episode's headline followed by a list of production credits. Subjectively examining the data, it is evident that the BART summary is often arguably more expressive and certainly more concise than the episode description:

```
DESCRIPTION: This episode’s headline: How’s Business?; Gene Breakthrough.
The guests include In Detroit: ROGER SMITH, Chairman, General Motors; In
New York: Akio morita, Chairman, Sony; GIOVANNI AGNELLI, Chairman, Fiat;
In Boston: Dr. DAVID NATHAN, Children’s Hospital; REPORTS FROM NEWSHOUR
CORRESPONDENTS: JUNE MASSELL. Byline: In New York: ROBERT MacNEIL, Executive
Editor; CHARLAYNE HUNTER-GAULT, Correspondent; In Washington: JIM LEHRER,
Associate Editor

SUMMARY: Duchenne’s is the most common form of muscular dystrophy. Scientists
have isolated the genes underlying two inherited diseases. The discovery of a
new gene could lead to a new way of treating cancer.
```

5.2 K-Means Clustering

Another core feature of the Graph Visualizer is the ability to cluster documents in real-time using the K-Means algorithm. Unsupervised methods like K-Means are, in general, very difficult to evaluate especially for natural language, since any categorization is almost entirely subjective. However, there are a few heuristics that can suggest the performance of a clustering system. One such metric is silhouette score, which is defined as the difference between the average distance of a point to other clusters and the average distance to other nodes within the same cluster, normalized by the maximum of the two:

$$\text{silhouette} = (b - a)/max(a, b) \text{ [3]}$$

This has the benefit of accounting for not only the degree of fittedness a node has to its cluster, but also its potential to belong to another. Calculating the average silhouette score of *all* nodes can therefore give a reasonable approximation of clustering ”confidence.”

	Sillhouette Score	
Number of total nodes	100	1,000
20 Clusters, all-mpnet-base-v2	0.05739843845	0.05073855445
10 Clusters, all-mpnet-base-v2	0.06408067793	0.06723831594
5 Clusters, all-mpnet-base-v2	0.09905090928	0.05979380012
10 Clusters, BOW	0.2737226973	0.2676438473
5 Clusters, BOW	**0.4076516027**	**0.7654857811**
10 Clusters, TF-IDF	0.08225996796	0.03597537923
5 Clusters, TF-IDF	0.08777493175	0.02421253796

Table 1: Results of clustering with different embedding methods and n clusters.

To determine the best clustering strategy for this program, it's also important to incorporate the fact that the clustering needs to be robust to different numbers of input nodes – the cluster quality should remain stable independent of how many files a user has uploaded.

The results [1], indicate an improved silhouette score when clustering on Bag of Words and TF-IDF features, rather than on context-embedded transformer representations. Since clustering is performed on summaries rather than entire documents, this is an understandable result; BART-generated summaries generally contain succinct lists of individual events, where context is not preserved nor relevant on a sentence-to-sentence level. This representation also comes with the added benefit of significantly faster performance of BOW compared to mnet-base, and, importantly for us, allows for a cluster representation that is conceptually distinct from the BERT-based topic modeling.

5.3 Topic Modeling

Similarly, topic modeling is another clustering feature of the visualization. Sticking with the primarily transformer-based architectures of the other tools, the Graph Visualizer performs topic modeling using BERTopic. BERTopic, unlike many other topic modeling strategies, uses a pipeline of steps including SBERT embedding, clustering, and TF-IDF weighting. Specifically, the visualizer implements BERTopic by pre-training it on a set of 2,258 AAPB NewsHour transcripts, then optionally expands the domain by re-training when new files are passed in.

As a further step, BERTopic allows for zero-shot topic modeling, which allows a user to employ their own domain knowledge of a dataset to define custom topics. Armed with these embedded manual topics, BERTopic trains two models on the data – one with manually defined topics and one without – and merges them into a single model. Especially for the axis topic representation, this can be extremely helpful for visualizing documents' relatedness to a given subject, with the caveat that a topic label is removed from the model if no documents of that topiccan be found.

Arguably, topic models are even more difficult to evaluate than general clus-

Config	default	CountVectorizer	CountVectorizer + ctfidf	CV + ctfidf + min_topic_size 20	CV + ctfidf + min_topic_size 30
Coherence	-0.7694668476	-0.9036572823	-0.20384128	-0.2882051108	-0.3487494821
Names	0 -1_the_to_and_of	0 -1_think_people_mr_going	0 -1_mr_years_just_new	0 -1_mr_people_think_thats	0 -1_think_people_mr_said
	1 0_the_to_and_of	1 0_iraq_war_think_jim	1 0_iraq_saddam_hussein_weapons	1 0_iraq_saddam_war_weapons	1 0_iraq_gwen_ifill_iraqi
	2 1_the_to_of_and	2 1_tax_think_mr_budget	2 1_tax_budget_cut_cuts	2 1_court_clinton_president_jim	2 1_iraq_saddam_war_weapons
	3 2_the_to_and_of	3 2_iraq_people_jim_think	3 2_kosovo_nato_milosevic_serbs	3 2_iraq_iraqi_baghdad_kerry	3 2_market_economy_going_rates
	4 3_the_and_to_in	4 3_kosovo_nato_think_milosevic	4 3_kerry_iraq_iraqi_fallujah	4 3_tax_budget_cut_billion	4 3_court_president_clinton_jim
	5 4_the_and_to_of	5 4_people_think_jim_hurricane	5 4_market_stock_markets_economy	5 4_kosovo_nato_milosevic_serbs	5 4_bush_campaign_party_kerry
...	...	...	...	...	...
Config	CV + ctfidf + min_topic_size 30	MMR diversity 0.015, 30 min_topic	MMR diversity 0.015, no min topic	TFIDF, max_df=0.85	TFIDF, max_df=0.99, min_df=0.6
Coherence	-0.3487494821	-0.5471053697	-0.1994792771	-0.3146514212	-0.6519895513
Names	0 -1_think_people_mr_said	0 -1_people_think_said_going	0 -1_mr_years_new_lehrer	0 -1_he_mr_his_lehrer	0 -1_warner_margaret_iraq_macneil
	1 0_iraq_gwen_ifill_iraqi	1 0_think_president_lehrer_jim	1 0_iraq_saddam_hussein_weapons	1 0_iraq_saddam_un_war	1 0_iraq_saddam_hussein_inspectors
	2 1_iraq_saddam_war_weapons	2 1_mr_macneil_woodruff_reagan	2 1_tax_budget_cut_cuts	2 1_tax_budget_bill_cut	2 1_education_macneil_deficit_programs
	3 2_market_economy_going_rates	3 2_iraq_people_gwen_iraqi	3 2_kerry_iraq_iraqi_fallujah	3 2_kosovo_nato_milosevic_serbs	3 2_iraq_kerry_iraqi_baghdad
	4 3_court_president_clinton_jim	4 3_market_think_economy_going	4 3_kosovo_nato_milosevic_serbs	4 3_hurricane_storm_orleans_water	4 3_kosovo_nato_serbs_serb
	5 4_bush_campaign_party_kerry	5 4_iraq_war_saddam_weapons	...	5 4_market_stock_paul_economy	...
	...	...			

Table 2: Topic modeling quantitative results

tering algorithms, as documents with shared topics may not neatly lie in a similar embedding space. One dimension that topic models are commonly evaluated in is (UMass) coherence score, which measures the similarity of the top words in a given topic cluster to measure how semantically "coherent" the clusters are.

This is a good start and gives us one of the only possible quantitative measures of the topic model's performance, but it fails to capture how well the model actually categorizes data points into topics – it is possible to have a model with extremely well-defined topic categories that still bins its data in a naive or incorrect way. In fact, an overconfident topic model may make a poor visualization if it generates sparse topic distributions, since each news document almost certainly contains multiple topics.

A working solution for this problem is twofold:

1. Tune hyperparameters of topic model for both effective clustering *and* correct chart distribution

2. Model probabilities using *approximate distribution* rather than taking topic probabilities directly. This method breaks a document into chunks, and categorizes each segment independently, taking the aggregate topic scores as the topic distribution for the entire document.

Approximate distributionsMethod 2, while powerful in capturing multiple diverse topic distributions in one file, areis extremely computationally expensive, since ithey multipliyes the number of topic model predictionsruns that need to be performed. To get around this problem, the graph visualizer performs topic modeling on "long summaries" – the concatenated BART chunk summaries stored before they were collapsed into a single one.

The model's ability to generate labels that are both comprehensible and not oversensitive can be measured using a custom metric integrating both the coherence score and the sparsity of the matrix (determined by its proportion of zeros). Running a grid search based on this criteria yields the best set of hyperparameters, which contain only a TFIDFVectorizer that clamps potential n-grams based on minimum and maximum document frequency scores; this provides the distributional benefits of the un-tuned (default representation) model

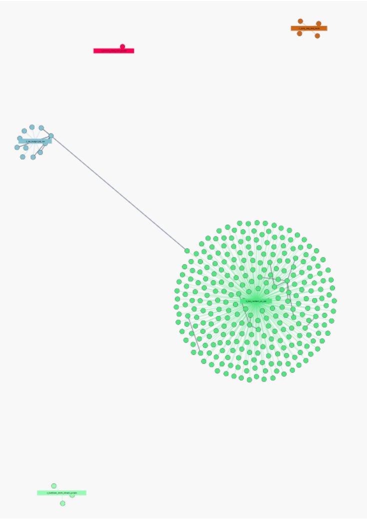


Figure 3: Many topic model configurations with high coherence scores form monolithic clusters

without producing as incomprehensible of label categories.

6 Visualization Evaluation

Evaluating a visualization platform is more often than not based on subjective measures of usefulness and aesthetics, since visualization is an inherently subjective task. For this project, I aimed to develop a quantitative metric for evaluating the platform that could also be applied to any document-set level visualization, using human performance as the score. [1]

First, taking a random sample of n nodes, run a prediction on each from a question generation model[2], leaving the target answer blank. This ensures true randomness in both question context and content, and will generate specific yet obscure questions.

For each question, measure the total ”clicks” required to ascertain an answer, including left mouse and Enter presses. This is compared against the baseline performance of O(n), where n is the total number of documents, which represents manually scanning through each document until the answer is encountered. In my case:

```
<pad> question: What did the former Attorney General say was an error
in the word "legal"?</s>

<pad> question: Who is the new ambassador to the United Nations?</s>
```

[1] Implemented in eval.py

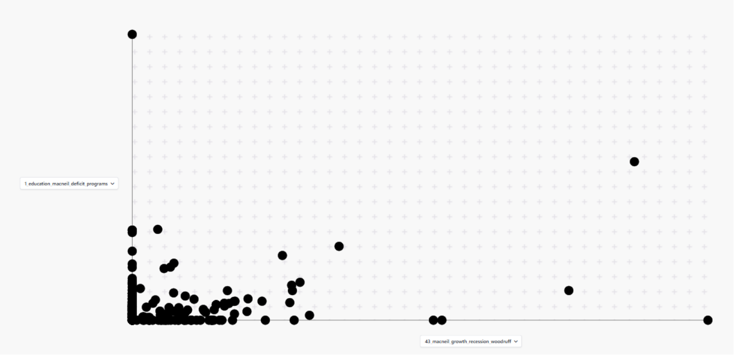


Figure 4: In the most expressive model, ”growth_recession” shows a relatively high rate of co-variance with ”education_programs”

```
<pad> question: What did the Philippines’s Rep. Solarz say was not presented to
support the election?</s>

<pad> question: What did the Vice President say he thought was a good speech?</s>

<pad> question: What was the name of the son of Robert Kennedy?</s>

<pad> question: What did the U.S. Secretary of State say about the Iraqi
government?</s>

<pad> question: What is the name of the U.N. High Commissioner for Refugees?</s>
<pad> question: What did Jerry West say about Michael Jordan’s decision to retire?</s>

<pad> question: What is the problem with Nicaragua?</s>

<pad> question: What is the Saudi leadership’s position on Iraq?</s>
```

The graph visualizer averaged 13 clicks for this task, with some deviation for more difficult questions. Since this metric is sensitive to the user, it is possible it may require fewer.

7 Limitations

As a dynamic physics simulation that runs entirely client-side in the browser, one of the main shortcomings of the program is efficiency. D3’s force-directed graph layout is fairly efficient at quickly rendering a small set of nodes, but it is very computationally expensive to load a large set of nodes and edges into memory (although once they are loaded by the client, performance issues are rare). This can lead to irritating load times when more files are uploaded, and therefore it is not recommended to load more than 300 nodes into the simulation at once.

Another computational hurdle comes with the many different types of modeling available. The program performs summarization at upload-time, since that is the most computationally expensive operation in the visualizer. However, because of this, file upload is slow, and should generally be done in bulk rather than in real-time.

References

[1] facebook/bart-large-cnn. https://huggingface.co/facebook/bart-large-cnn. Accessed: 2024-05-06.

[2] mrm8488/t5-base-finetuned-question-generation-ap. https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap. Accessed: 2024-05-06.

[3] silhouette_score. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html. Accessed: 2024-05-06.

[4] Matthew Berger, Katherine McDonough, and Lee M Seversky. cite2vec: Citation-driven document exploration via word embeddings. *IEEE transactions on visualization and computer graphics*, 23(1):691–700, 2016.

[5] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

[6] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. ivisclustering: An interactive visual document clustering via topic modeling. In *Computer graphics forum*, volume 31, pages 1155–1164. Wiley Online Library, 2012.

[7] Hendrik Strobelt, Daniela Oelke, Christian Rohrdantz, Andreas Stoffel, Daniel A Keim, and Oliver Deussen. Document cards: A top trumps visualization for documents. *IEEE transactions on visualization and computer graphics*, 15(6):1145–1152, 2009.

[8] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2024.