

Narrative Trails: A Method for Coherent Storyline Extraction via Maximum Capacity Path Optimization

Fausto German^{1,*}, Brian Keith² and Chris North¹

¹Virginia Tech, Blacksburg, Virginia 24061, USA

²Universidad Católica del Norte, Av. Angamos 0610, Antofagasta, 1270709, Chile

Abstract

Traditional information retrieval is primarily concerned with finding relevant information from large datasets without imposing a structure within the retrieved pieces of data. However, structuring information in the form of narratives—ordered sets of documents that form coherent storylines—allows us to identify, interpret, and share insights about the connections and relationships between the ideas presented in the data. Despite their significance, current approaches for algorithmically extracting storylines from data are scarce, with existing methods primarily relying on intricate word-based heuristics and auxiliary document structures. Moreover, many of these methods are difficult to scale to large datasets and general contexts, as they are designed to extract storylines for narrow tasks. In this paper, we propose Narrative Trails, an efficient, general-purpose method for extracting coherent storylines in large text corpora. Specifically, our method uses the semantic-level information embedded in the latent space of deep learning models to build a sparse coherence graph and extract narratives that maximize the minimum coherence of the storylines. By quantitatively evaluating our proposed methods on two distinct narrative extraction tasks, we show the generalizability and scalability of Narrative Trails in multiple contexts while also simplifying the extraction pipeline. The code for our algorithm, evaluations, and examples are available at <https://github.com/faustogerman/narrative-trails>

Keywords

Narrative Extraction, Coherence Graph, Information Extraction, Information Retrieval, Sensemaking

1. Introduction

In the last couple of decades, the fields of data science and data analytics have seen significant growth, helping people make sense of large, complex, and often interwoven data. A common task in the sensemaking process is to structure data in a format that aids analysis and information retrieval for downstream tasks [1]. For example, structuring information in the form of narratives can help scientists communicate advanced ideas to the general public [2, 3] and can aid with finding information in collaborative settings [4]. That is, narratives serve as tools for structuring complex datasets into coherent, manageable units that facilitate more effective communication and understanding of the information, ultimately reducing the cognitive load needed to make sense of information [5]. By organizing disparate data points into narrative structures, we enable people to identify underlying patterns, connections, and themes that might not be immediately evident by the data. For instance, placing documents in a sequential storyline may help a student researching the relationship between “computer vision” (CV) and “natural language processing” (NLP) to discover that “image captioning” and “visual question answering” bridge the concepts of CV and NLP.

However, despite the importance of narrative extraction from data, efficient algorithmic approaches are scarce, with current methods primarily relying on intricate word-based heuristics [6] and linear programming formulations [5, 7] with limited scalability and versatility. To solve some of the scalability and availability issues of current methods, in this work we propose Narrative Trails, an algorithm for extracting coherent storylines from text documents. This helps to relate potentially disconnected ideas

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Litvak (eds.): *Proceedings of the Text2Story’25 Workshop, Lucca (Italy), 10-April-2025*

*Corresponding author.

✉ fgermanj@vt.edu (F. German)

🌐 <https://faustogerman.com> (F. German)

🆔 0009-0005-0954-4578 (F. German)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

and extract information from large datasets. To achieve this, we approach the narrative extraction problem as a maximum capacity path optimization problem. Specifically, we utilize Dijkstra’s algorithm with a MaxiMin objective to identify k distinct paths that maximize the minimum coherence between documents in a sparse coherence graph representation extracted from the dataset. Following the metaphor of Narrative Maps [7], the Narrative Trails algorithm is analogous to route maps for hiking trails, which often provide a multiple adjacent path between a starting point and a destination.

To assess the performance of our proposed method, we present a quantitative comparison of the Narrative Trails algorithm against a random sampling, shortest simple path, and a simplified version of the Narrative Maps extraction method on two distinct tasks across four datasets. Our analyses show that the proposed approach has a lower computational cost and better performance in terms of narrative coherence.

In this paper, we make the following contributions to computational narrative extraction: (1) **Approach**: We provide a more abstractive approach to narrative extraction, focused primarily on the abstract semantic relationships between documents in a dataset; (2) **Algorithm**: We describe an efficient algorithm that merges dimensionality reduction and path optimization for coherent storyline extraction from large datasets; and (3) **Extensibility**: We provide a repository with details of our algorithm that can be used to reproduce our results or to extend our methods to other contexts beyond text.

In the next section, we review related work on narrative extraction. Section 3 formalizes the Narrative Trails extraction method and its core components. Sections 4 and 5 present and analyze our evaluation results across multiple extraction tasks. Finally, we discuss the limitations of our work and potential lines of future research in Section 6, followed by conclusions in Section 7.

2. Related Work

Computational narrative extraction lies at the intersection of artificial intelligence, natural language processing (NLP), and combinatorial optimization. The interdisciplinary nature of computational narratives makes them useful for knowledge discovery and information synthesis from large sets of data, as they allow the extraction of structured content from unstructured content. Moreover, recent developments in NLP and deep learning models have driven computational narrative extraction to a notable area of scholarly discussion [8].

Many frameworks, models, and algorithms have been developed for computational narrative extraction, including linear sequences [5, 6] or timelines [9, 10], parallel stories [11, 12], and directed acyclic graphs [7, 13, 14]. Each of these revolves around the idea of storylines that weave narrative elements—sentences, documents, or clusters of documents—into coherent sequences of events. For instance, in the Narrative Maps [7] and Metro Maps [14] approaches, the authors build directed acyclic graphs that interconnect multiple storylines into single narratives with potentially many starting and ending events around particular subjects. These approaches aim to extract the underlying graph structure of documents through optimization techniques that prioritize the coherence of the storylines while adhering to the global structural constraints of the narratives.

Similarly, the work of Xu et al. [11] aims to build multiple timelines or storylines parallel to one another that share similarities or revolve around a complex topic. On the other hand, The newsLens algorithm [12] builds multiple parallel timelines across an entire dataset that may or may not share common themes. These methods underscore the importance of narrative frameworks in providing a multifaceted view of complex topics, allowing for a richer, more nuanced understanding of events and their interconnections while acknowledging the separation in ideas between each storyline.

Finally, in the Connect the Dots approach [5], the authors focus on extracting singular sequences of events that together form a narrative chain between two fixed endpoints. This method only considers one storyline at a time, displaying a linear narrative progression of the most relevant documents from a defined start to an end. They achieve this by optimizing a set of word-based linear programming constraints that maximize the coherence of the chain.

In our work, we also focus on extracting singular narrative chains. However, we focus on an

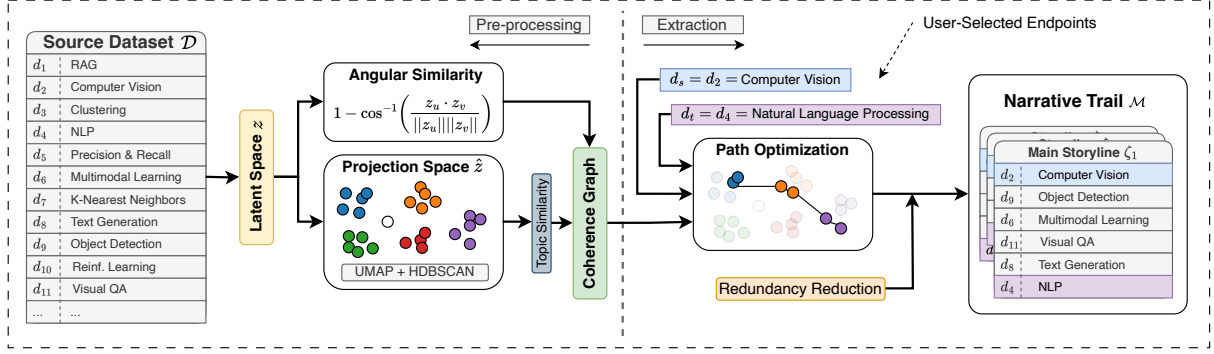


Figure 1: Narrative Trails extraction pipeline. Given two user-selected documents, the algorithm finds storylines that connect them with maximum capacity for coherence.

abstractive approach based on the semantics of a piece of text rather than the activations of individual words within the text and their exact appearance across the narrative. This allows us to extract chains between documents that may not use the exact wording but nevertheless share similar themes. Moreover, given enough data, our abstractive approach can find smooth transitions between unrelated source and target documents.

3. Methodology: Narrative Trails

In this section, we introduce the formal definition of a Narrative Trail, which serves as the foundation for the structure of our narrative extraction methods.

Definition 1 (Narrative Trail). Let $G = (V, E, w_f)$ be a weighted, directed graph built from a set of documents $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$, with each node $v \in V$ having an associated document $d_v \in \mathcal{D}$ and edges $(u, v) \in E$ weighted through some coherence function $w_f(u, v) = f(d_u, d_v) \geq 0$. A Narrative Trail \mathcal{M} is a collection of storylines $\{\zeta_1, \zeta_2, \dots, \zeta_k\}$ in G , where each storyline $\zeta_k = \{d_s, d_1^k, d_2^k, \dots, d_l^k, d_t\}$ is an ordered sequence of documents that maximize the minimum edge weight $w_f(u, v)$ for all $(u, v) \in E$ to connect user-selected endpoints $s, t \in V$. Documents $d_1^k, d_2^k, \dots, d_l^k$ are unique to their storylines.

Our algorithm is a simplification of Narrative Maps [7] for the single-pair narrative extraction case with up to k distinct storylines between the source and target documents. However, unlike previous work, Narrative Trails does not rely on linear programming to extract storylines. Instead, the algorithm focuses on the latent spaces learned by deep learning models for path optimization, with the coherence graph structure and its maximum spanning tree being natural representations of the connection between documents in this space. More concretely, we divide the algorithm into projection space construction, coherence graph representation, and path optimization. Figure 1 provides an overview of the extraction pipeline and the following subsections provide more details about each step.

3.1. Projection Space Construction

Our narrative extraction algorithm relies on a low-dimensional data representation constructed from the latent space learned by a deep learning model to discover topics for each document, which we achieve through a neural topic modeling method similar to BERTopic [15]. More specifically, we use the text-embedding-3-small embedding model from OpenAI’s API service. While OpenAI’s second-generation model, text-embedding-ada-002, had previously shown great performance in various NLP tasks [16, 17], their latest embedding models outperform the previous models in benchmarks for English tasks and multi-language retrieval [18]. We also note that other embedding models can produce similar results. For instance, the all-mpnet-base-v2 model from the Sentence Transformers library [19] has shown state-of-the-art performance in multiple embedding tasks such as semantic similarity and retrieval [20, 21], providing an excellent balance between speed and performance for embedding

extraction. However, it is limited to a smaller context length of 384 tokens, restricting the amount of information that can be captured within the embedding representations. OpenAI’s models do not have this limitation, as they allow API calls with thousands of tokens at a time.

We then use a combination of the Uniform Manifold Approximation and Projection (UMAP) algorithm [22] and HDBSCAN [23] to project the data into two dimensions and assign clusters to each document. UMAP has been shown to outperform other dimensionality reduction techniques such as t-SNE [24] and PCA by preserving more of the local and global structure of the embeddings [25, 26]. Since HDBSCAN is a density-based clustering algorithm, UMAP also serves as a complementary technique by creating low-dimensional projections where documents are more densely grouped, thereby mitigating the effects of the curse of dimensionality [27]. Additionally, HDBSCAN’s soft-clustering feature allows us to obtain cluster probability distribution vectors, which capture the probability of each document belonging to each of the discovered clusters. Comparing the topic distributions of two documents then allows us to quantify their topic similarity [7], which is an important component for ensuring the extracted storylines follow smooth topic transitions.

3.2. Coherence Graph Representation

Narrative Trails builds storylines by connecting documents based on the content similarity encoded in their embeddings. A naive approach to constructing the storylines is to let some storyline ζ_i be the shortest Euclidean path in the latent space between the source and target documents. However, using spatial information from the latent space alone does not provide enough structure for narrative extraction. This is because as the size of the dataset approaches infinity, finding a path between two embeddings using only their Euclidean distance would equate to finding a path of shortest distance in \mathbb{R}^n , which would resemble a straight line between the source and target points [28]. However, deep learning models do not learn globally linear embedding spaces. Therefore, a straight line through the latent space may not capture the semantics of the documents well enough to define a coherent narrative. Instead, we need to define a quantity that measures how plausible it is for two documents to be connected in a storyline based on the semantics encoded in the latent space. That is, we need to define a measure for document pairwise coherence.

3.2.1. Base Coherence

Building on the premise that documents within a narrative should share content and context similarity [5, 29, 30], prior work [7] defines the coherence $\theta(d_u, d_v)$ between two documents d_u and d_v as the geometric mean of their angular similarity in the high-dimensional embedding space and their topic similarity in the low-dimensional projection space. Formally, the coherence is defined as:

$$\theta(d_u, d_v) = \sqrt{S(z_u, z_v)T(\hat{z}_u, \hat{z}_v)} = \sqrt{(1 - \arccos(\cos_sim(z_u, z_v))/\pi) (1 - \text{JSD}(\hat{z}_u, \hat{z}_v))} \quad (1)$$

where $z_u, z_v \in \mathbb{R}^n$ are the high-dimensional embeddings in the latent space for documents d_u and d_v , and $\hat{z}_u, \hat{z}_v \in \mathbb{R}^m$ are their low-dimensional projections (with $m \ll n$). The angular similarity $S(z_u, z_v)$ maps the angle between the embeddings to a similarity measure in the interval $[0, 1]$, and the topic similarity $T(\hat{z}_u, \hat{z}_v)$ is defined using the Jensen-Shannon divergence (JSD) between the cluster membership distributions of the documents in the low-dimensional space. We obtain the cluster membership distribution vectors from HDBSCAN during the projection space construction step. Thus, two documents are considered highly coherent with respect to one another if they exhibit *both* high content similarity from $S(z_u, z_v)$ and high topic similarity from $T(\hat{z}_u, \hat{z}_v)$.

3.2.2. Sparse Coherence

We note that the base coherence results in a complete, undirected graph $G = (V, E, w_\theta)$, where the weights w_θ are defined by the function $\theta(d_u, d_v)$. However, we can induce sparsity into G by leveraging properties of its maximum spanning tree MaxST_G , which is the inverse of the minimum spanning tree

[31]. Specifically, we observe that in an undirected graph, any s - t path in the maximum spanning tree is also a path that maximizes the minimum edge weight between s and t in the original graph G [32]. This follows from the fact that MaxST_G is constructed by selecting the highest-weight edges while maintaining connectivity [31], which ensures that for any two nodes, the weakest edge along their path is as strong as possible.

Given the equivalence between the storylines in the original graph and those in its maximum spanning tree, a tree-based approach serves as an additional optimization strategy by reducing the search space from the source node to the target during the storyline extraction phase. However, while it is possible to use MaxST_G directly to identify the storylines ζ_i , trees inherently provide only a single path between any two vertices. This limitation conflicts with the requirement to extract multiple storylines, as outlined in Definition 1. Consequently, this structure may be too restrictive in scenarios where multiple storylines are necessary. For example, extracting the top- k distinct storylines can aid in sensemaking, as each storyline may reveal a different perspective on the relationship between the endpoints. To address this, we extend the base coherence function by introducing *sparse coherence*, which is defined as follows to balance sparsity with the number of possible storylines between two documents:

$$\vartheta(d_u, d_v) = \mathbb{1}[u \neq v \text{ and } \theta(d_u, d_v) \geq \tau\omega] \theta(d_u, d_v) \quad (2)$$

Where ω is the bottleneck edge weight of MaxST_G . The parameter $\tau \geq 0$ scales the bottleneck weight to set a hard cutoff on the minimum coherence between any two documents. It follows from Equation 2 and Kruskal’s algorithm [31] for maximum spanning trees that if $\tau > 1$, MaxST_G becomes disconnected and therefore the resulting sparse graph *may* also become disconnected. This is because a value of $\tau > 1$ has the effect of raising ω past the bottleneck weight that holds together MaxST_G . In contrast, if $\tau \leq 1$, the tree remains connected, and we can construct at least one storyline between two nodes in G .

3.2.3. Incorporating Task-Specific Information

While so far we have only discussed undirected graphs, Equation 2 allows us to model complex constraints through task-specific information that induce explicit directionality to the storylines. For instance, time dependencies between documents can be enforced by refining the sparse coherence definition to include an additional condition $\gamma(d_u, d_v)$ within the indicator function. The function $\gamma(d_u, d_v)$ returns `true` if and only if the date of document d_v is later than that of document d_u , ensuring that the extracted storylines follow chronological order. To that end, the final step in the Coherence Graph Representation is to construct a weighted, directed coherence graph $G_\vartheta = (V, E, w_\vartheta)$, where nodes $v \in V$ represent the documents in our dataset and edges $(u, v) \in E$ with weights $w_\vartheta(u, v)$ are formed based on the sparse coherence $\vartheta(d_u, d_v)$ between documents. Specifically, an edge from node u to node v exists if and only if $w_\vartheta(u, v) > 0$.

3.3. Path Optimization

Recall that our objective is to extract a collection of k distinct storylines that maximize the minimum edge weight to connect a source document d_s to a target document d_t in a weighted directed graph G . By letting G equal the sparse coherence graph G_ϑ constructed in section 3.2, the extracted storylines then aim to optimize the minimum edge value or, equivalently, maximize the minimum coherence required to connect a predefined source document d_s to a target document d_t .

The problem outlined mirrors the Maximum Capacity Path problem [32], aiming to maximize the minimum edge weight within a graph. While linear-time algorithms exist for solving this problem [33], they require additional constraints or assumptions about the graph, such as undirected graphs with distinct edges or weights in \mathbb{N} [34]. Our coherence graph does not meet either of those requirements since it is a directed graph with real-valued edge weights from the sparse coherence between nodes. Given these constraints, we repurposed Dijkstra’s algorithm [35] with a single-pair path-finding task and a maximin objective for storyline extraction. In particular, this version of Dijkstra’s algorithm

updates the tentative score of a node by taking the minimum between the current path’s minimum edge weight and the edge weight leading to the node.

3.3.1. Extracting k Distinct Chains

To identify the top k distinct storylines from the source node s to the target node t , we modify Dijkstra’s algorithm to exclude nodes that have already been included in previously discovered paths. Specifically, after each execution of the algorithm, we record the set of nodes $V_p = \bigcup_{i=1}^k (\zeta_i \setminus \{d_s, d_t\})$ representing the nodes contained in each previously discovered path ζ_i and update the graph to ignore these nodes in subsequent extractions. That is, we iteratively run the modified algorithm k times, each time operating on the updated graph $G'_g = (V \setminus V_p, E')$, where E' includes only edges between the remaining nodes. By effectively “hiding” these nodes in subsequent extractions, we prevent them from being part of any new storylines.

3.3.2. Redundancy Reduction

An inherent property of spanning trees is that the paths between vertices may not always be the most direct, often resulting in longer routes through the graph [31]. This occurs because eliminating cycles reduces the number of possible shortcuts between nodes. In our pipeline, this means that the extracted storylines can sometimes be excessively long or include redundant documents. To address this issue, we implement a fast post-processing step that removes redundant documents from the storylines.

For each extracted storyline ζ_i , we examine consecutive triplets of documents (A, B, C) and calculate the sparse coherence values between them. We let R be the geometric mean of the base coherence values $\vartheta(A, B)$ and $\vartheta(B, C)$. Since R is, by definition, greater than or equal to the minimum edge weight ω_ζ in the storyline, we only check whether $\vartheta(A, C) \geq R - \delta$ if the edge (A, C) exists in the sparse coherence graph, where δ is a redundancy threshold parameter. If true, we consider document B to be redundant and remove it from the storyline. This process creates a shortcut from A to C without significantly compromising the overall coherence, resulting in potentially more concise storylines.

4. Experiments & Evaluations

In this section, we implement our proposed Narrative Trails pipeline and evaluate its performance on two distinct narrative extraction tasks to address the following questions: (RQ1) How well does Narrative Trails align with human-derived shortest semantic paths? and (RQ2) How do the storylines extracted by Narrative Trails compare to those extracted by the current state-of-the-art method? Our goal with these evaluations is to demonstrate the generalizability of Narrative Trails across multiple domains and tasks, as well as its adaptability to various sensemaking scenarios. Additionally, we illustrate how the flexibility of our sparse coherence definition allows us to incorporate task-specific information and constraints into the extraction pipeline.

4.1. Evaluation Metrics

To evaluate the intrinsic quality of the storylines, we (1) measure the minimum coherence within each storyline to verify how well our method maximizes the weakest link in the chain and (2) calculate the geometric mean of the coherence values of the edges in the storylines. We call this the “reliability” of the storyline as it indicates the likelihood that the documents collaboratively form a coherent storyline.

To evaluate the similarity between two storylines of potentially different lengths, we first identify the Dynamic Time Warping (DTW) path [36] between them using the Euclidean distance between the low-dimensional embeddings of their documents as the matching metric and normalize by the number of matches in the path (referred to hereafter as nDTW Distance), then compute the average pairwise cosine similarities between those embeddings along the resulting DTW path (referred to hereafter as DTW Similarity). Dynamic Time Warping is commonly used in time series search [37] and as a metric

for curve similarity [38]. In this context, the DTW metrics measure the semantic alignment between storylines, which we represent as curves in the low-dimensional projections. This allows us to quantify the similarity between storylines of different lengths that share related but distinct documents.

4.2. Experimental Setup

We used four datasets and two tasks to evaluate our proposed methods. To answer RQ1, we use a subset of human-derived paths over the Wikipedia network through the Wikispeedia game [39]. In this game, users are tasked with finding a path between two Wikipedia pages in as few clicks as possible. Although the objective of the game—to find a shortest path—is slightly different than the goal of our proposed methods—to find a path of maximum semantic capacity—the paths extracted by humans in the Wikispeedia game have been shown to encode context about how humans perceive and explore information, especially in relation to the semantics of the network and the assigned goal page [40]. To that end, we select a subset of 10,607 finished paths from the dataset (covering 3,928 Wikipedia pages) with lengths between 7 and 20 pages per path and no back links to use as a ground truth dataset.

To answer RQ2, we use a collection of 540 news articles related to the COVID-19 pandemic and the 2021 Cuban protests [41], 840 randomly sampled research articles from the VisPub dataset [42], and 1,140 randomly sampled research articles related to machine learning and AI from the AMiner dataset [43]. These datasets feature various subtopics under a single main topic, allowing for a more focused evaluation of our experiments. In addition, we selected the AMiner and VisPub subsets at random to minimize any bias in the subtopic distributions that could provide an advantage to any of the algorithms. We implemented the Narrative Maps algorithm as a baseline by removing the coverage constraint from its linear programming formulation and setting the expected length for the main storyline extraction to 12 documents, ensuring a fair comparison with Narrative Trails.

Since OpenAI’s `text-embedding-3-small` model provides 1536-dimensional embeddings, we project them to a 48 dimensions using UMAP before clustering with HDBSCAN. In all experiments, we implemented the proposed Narrative Trails algorithm and used the default values of $\tau = 1$ and $\delta = 0.2$ for the sparse coherence formulation and redundancy reduction, respectively. The choice of $\tau = 1$ follows from Equation 2, ensuring that the graph remains connected by the critical edge weight ω . Additionally, our empirical analysis shows an average critical coherence value of 0.58 ± 0.104 across datasets. Based on this, we set $\delta = 0.2$ to balance coherence and flexibility, preventing the storylines from becoming overly incoherent while allowing some variation in the documents considered redundant. In the Wikispeedia experiments, we incorporated the directed edges of the Wikipedia network as an additional task-specific constraint within the sparse coherence formulation. When comparing against Narrative Maps, we used the publication dates of the articles to enforce directionality on the edges, as detailed in section 3.2.3. Additionally, we benchmarked Narrative Trails against a random sampling method and a shortest simple path algorithm across all experimental setups.

5. Results

5.1. Alignment with Human-Derived Paths

For our Wikispeedia evaluations, we extracted the top $k = 3$ distinct paths between the source and target documents in each of the ground truth human-extracted storylines. We average the scores of the top- k storylines to provide a sense of the cumulative extraction quality. Table 1 summarizes the results of this experiment. In most cases, Narrative Trails outperforms the random sampling and simple shortest path algorithms. However, in the case of nDTW Distance, the shortest simple paths outperform our methods since it more closely models the underlying task of the Wikispeedia game by node count.

Evaluations of how humans navigate information networks using a directed *st*-task demonstrate that users typically visit hub nodes—Wikipedia pages with many incoming and outgoing links—before zeroing in on the target document [40]. Building on this observation, we investigated whether Narrative Trails could emulate similar behavior by multiplying each node’s base coherence by their closeness centrality

Table 1

Comparison of absolute coherence and reliability, along with DTW similarity and distance for the top- k extracted storylines between Narrative Trails, Narrative Trails with closeness centrality (CC), and shortest simple path using the human-derived paths from the Wikipedia dataset as ground truth.

Method	Min. Coherence			Reliability			DTW Similarity			nDTW Distance		
	$k = 1$	$k = 2$	$k = 3$	$k = 1$	$k = 2$	$k = 3$	$k = 1$	$k = 2$	$k = 3$	$k = 1$	$k = 2$	$k = 3$
Wikispeedia	0.419	—	—	0.609	—	—	—	—	—	—	—	—
Random Points	0.320	0.321	0.322	0.454	0.455	0.456	0.347	0.347	0.347	2.200	2.201	2.200
Shortest Path	0.558	0.560	0.563	0.614	0.615	0.620	0.742	0.742	0.746	0.967	0.978	0.971
Narrative Trails	0.709	0.704	0.704	0.776	0.769	0.767	0.788	0.785	0.787	1.029	1.049	1.063
Redundancy Reduced	0.668	0.667	0.669	0.760	0.756	0.755	0.769 [†]	0.768	0.771 [†]	1.055	1.076	1.088
Narrative Trails (CC)	0.640	0.631	0.630	0.753	0.748	0.746	0.777	0.778	0.766 [†]	1.029	1.049	1.093
Redundancy Reduced (CC)	0.630	0.625	0.624	0.737	0.735	0.734	0.759	0.761 [†]	0.751	1.065	1.079	1.117

Table 2

Comparison of absolute coherence and reliability, along with DTW similarity and distance for the top- k extracted storylines between Narrative Trails and the shortest simple path using the Narrative Maps algorithm as baseline.

Method	Min. Coherence			Reliability			DTW Similarity			nDTW Distance		
	News	VisPub	AMnr.	News	VisPub	AMnr.	News	VisPub	AMnr.	News	VisPub	AMnr.
Narrative Maps	0.499	0.554	0.502	0.702	0.677	0.629	—	—	—	—	—	—
Random Sample	0.343	0.412	0.357	0.492	0.577	0.512	0.621	0.337	0.278	2.466	1.397	1.427
Shortest Path	0.557	0.743	0.635	0.593	0.753	0.644	0.363	0.461	0.188	1.001	0.991	1.108
Narrative Trails	0.689	0.784	0.736	0.786	0.800	0.764	0.872	0.616	0.556	0.762	0.915[†]	0.962
Redundancy Reduced	0.638	0.756	0.691	0.739	0.777	0.724	0.845	0.570 [†]	0.455	0.825	0.946 [†]	1.025 [†]

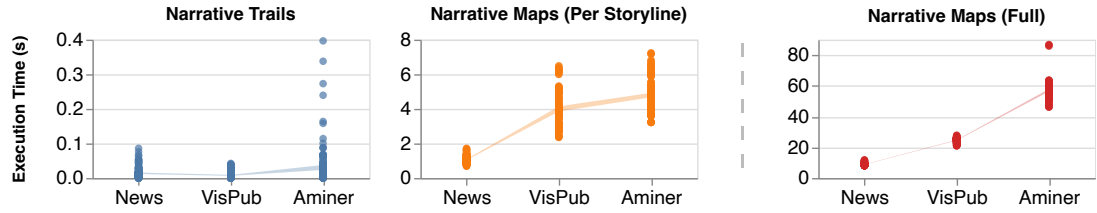


Figure 2: Comparison of extraction execution time per storyline between Narrative Trails and Narrative Maps. The diagonal lines indicate the error bands for execution time across different datasets for each algorithm.

[44] score along outgoing edges. As illustrated in Table 1, Narrative Trails with closeness centrality becomes the second-best performer in most cases of DTW Similarity (underlined), demonstrating the flexibility of our algorithm to approximate human sensemaking.

5.2. Comparison with Narrative Maps

For our comparison with Narrative Maps, we extracted the top storyline with Narrative Trails and the equivalent main storyline with Narrative Maps for 50 randomly sampled source-target pairs from each dataset. Table 2 summarizes the results of our experiments with Narrative Trails, using Narrative Maps as a baseline algorithm. In all datasets, Narrative Trails extracts storylines with higher coherence and reliability. Moreover, when compared against the random sampling and shortest simple paths methods with the results from Narrative Maps as ground truth, Narrative Trails extracts storylines that semantically align better with the state-of-the-art Narrative Maps algorithm.

Similar to our coherence evaluation, we measured the execution time of Narrative Trails and Narrative Maps on all datasets. Since each dataset had a different number of documents, we could get a sense for how well each of the algorithms scales during the storyline extraction phase. Specifically, we measured the time it takes the algorithms to extract storylines and disregarded the time required to construct the projection space and coherence graphs. These excluded steps are considered preprocessing tasks in both algorithms, involving non-critical and cacheable operations from the end user’s perspective.

Our simplified narrative extraction pipeline substantially speeds up our algorithm’s performance compared to Narrative Maps. Figure 2 shows the average execution time per extracted storyline for both

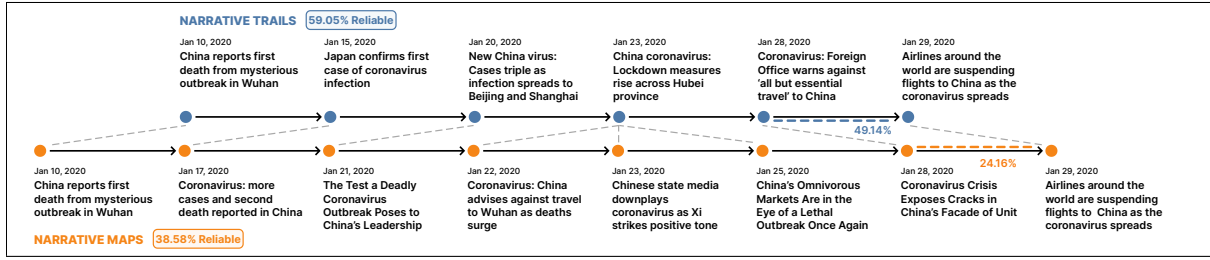


Figure 3: Storylines about the COVID-19 pandemic’s impact on global flights in January 2020, extracted from a collection of news articles using Narrative Trails (blue) and Narrative Maps (orange). The dashed gray lines represent the DTW matching between the storylines, and the dashed colored lines are the weakest links.

algorithms on the News Articles, VisPub, and AMiner datasets. We also note that since Narrative Maps is required to extract many storylines within the same narrative, we have divided the total extraction time by the number of storylines extracted (orange chart) to maintain a fair evaluation. However, we also show the execution times for the full extraction of each narrative map (red chart). These results demonstrate the efficiency of Narrative Trails when compared to Narrative Maps, especially for datasets with thousands of documents.

Lastly, Figure 3 showcases example storylines extracted by both algorithms to connect the first reported death from the SARS-CoV-2 virus to airlines worldwide canceling flights to China. While both storylines track the development and international response to the coronavirus outbreak, Narrative Trails emphasizes the specific developments within the health crisis that prompted an immediate response from airlines, making the storyline focused on the cause (the spread of the virus) and the effect (the suspension of flights).

5.3. Statistical Analyses

In most cases, our base Narrative Trails algorithm, without redundancy reduction, shows statistically significant differences across all metrics compared to other methods. However, in some instances—specifically when compared to shortest paths—the results for DTW Similarity and Distance for the redundancy reduced Narrative Trails are not statistically significant. In Tables 1 and 2, we mark such cases with †. This suggests that our redundancy reduction algorithm causes the maximum capacity path to approximate the shortest simple path between the source and target documents. Additional details on the statistical analyses are available in the linked GitHub repository.

6. Limitations and Future Work

Evaluation Limitations We note that the evaluations with the user-extracted Wikispeedia paths as a ground-truth dataset may be difficult to interpret, as the task of the game differs slightly from the objective of narrative extraction. Thus, performing a user study or an expert-based analysis of our proposed algorithm could help to verify and contextualize its broader usability and alignment to human perception of storyline coherence.

Similarly, we do not include methods such as Connect the Dots [5] and newsLens [12] in our evaluations. This is because the highly focused nature of these algorithms and their code availability make it difficult to incorporate them into our generalized evaluation pipeline. Additionally, our focus on single storylines in the evaluations against Narrative Maps, as opposed to the more complex narrative structures possible with other algorithms, points to future work to develop general narrative evaluation metrics that can take into account multiple interconnected storylines.

Lastly, while our evaluations report coherence—which our algorithm optimizes—as a key metric, the additional DTW metrics used in our evaluations do not rely on our optimized coherence function. Instead, DTW measures sequence similarity based on structural alignment, providing an independent assessment of storyline similarity against the Wikispeedia and Narrative Maps baselines.

Comparison Issues with Narrative Maps For a fair comparison against our methods, we removed the coverage constraint from the linear programming formulation of the Narrative Maps algorithm. The reason for this removal was twofold: (1) Narrative Trails does not include coverage constraints in its formulation, focusing solely on coherence; and (2) adding the coverage constraints is a bottleneck of the Narrative Maps extraction algorithm when the number of clusters is high, as in our evaluations, which would inflate the execution times unfairly. In practical terms, removing these constraints is equivalent to requiring a minimum average coverage of 0% in the Narrative Maps extraction algorithm. In this context, we note that removing the coverage constraints can lead Narrative Maps to focus on a single topical cluster as it no longer needs to address the diversity requirements in topical coverage, which reduces its overall performance.

Another relevant point of comparison is that, in Narrative Maps, the extraction process can be guided by the user through semantic interactions that can directly alter its linear programming constraints. This approach provides a direct way to guide the narrative extraction and generate relevant narratives for the user. In contrast, Narrative Trails requires constraints and task-specific information to be encoded during pre-processing. Future research could explore semantic interaction models that enable users to dynamically modify and update the sparse coherence graph during extraction. For instance, deep-learning-based search agents could encode human-controllable constraints directly into its learned parameters, offering a balance between speed, accuracy, and user-defined narrative requirements as part of the semantic interaction pipeline.

Potential Extensions to Image Data Lastly, we note that the Narrative Trails algorithm is data and model-agnostic and can be extended to any context where embeddings are available. Beyond text, a trivial extension of this algorithm could extract “concept narratives” from image data that transition the concepts in a source image to the concepts of a target image. Applications of this extension include guided storyboarding and multi-modal narrative generation. However, these scenarios require considering the semantics of the data, as well as the representation power of the deep learning model used for embedding extraction, information retrieval, and similarity search in the context of computer vision and multi-modal learning.

7. Conclusions

In this paper, we presented Narrative Trails, a method for storyline extraction from large datasets with task-specific graph structures. This method addresses the limitations of current algorithms by providing an abstractive approach to narrative extraction. More specifically, we leverage the representational power of deep learning models to capture the semantics of data to form storylines. Our main insight stems from the parallels between our definition of coherent storylines and the maximum capacity path problem. The results from our experiments on human-derived paths from the Wikispeedia dataset and the comparative study with Narrative Maps demonstrate the ability of our presented algorithm to not only generalize to task-specific domains, but also to extract storylines with high coherence. Moreover, our simplified graph construction and extraction pipeline improves the overall time complexity over current methods, opening the door to future methods where extraction time is critical.

Our narrative extraction algorithm opens new avenues for future research, including contextualized global coherence and semantic interaction. One key avenue lies in improving our definition of coherence with global metrics that better guide the extraction process and regulate storyline length. Additionally, integrating deep-learning-based search agents with human-controllable constraints could improve the balance between speed, accuracy, and user-defined requirements. These improvements could further generalize our approach, broadening its applicability to context-specific domains and tasks.

Despite its limitations, our approach marks a significant contribution to computational narrative extraction, offering a more general method that simplifies the storyline extraction pipeline. Additionally, the data and model-agnostic nature of Narrative Trails opens the doors to extracting narratives from diverse datasets, including image data, thus expanding the method’s applicability and impact.

Acknowledgments

Brian Keith is supported by Project 202311010033-VRIDT-UCN.

References

- [1] P. Pirolli, S. Card, The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis, in: *Proceedings of International Conference on Intelligence Analysis*, volume 5, McLean, VA, USA, 2005, pp. 2–4.
- [2] Y. Yang, J. E. Hobbs, The power of stories: Narratives and information framing effects in science communication, *American Journal of Agricultural Economics* 102 (2020) 1271–1296. doi:<https://doi.org/10.1002/ajae.12078>.
- [3] M. W. Kreuter, M. C. Green, J. N. Cappella, M. D. Slater, M. E. Wise, D. Storey, E. M. Clark, D. J. O’Keefe, D. O. Erwin, K. Holmes, L. J. Hinyard, T. Houston, S. Woolley, Narrative communication in cancer prevention and control: a framework to guide research and application, *Annals of Behavioral Medicine: A Publication of the Society of Behavioral Medicine* 33 (2007) 221–235. doi:[10.1007/BF02879904](https://doi.org/10.1007/BF02879904).
- [4] A. Karunakaran, M. Reddy, The role of narratives in collaborative information seeking, in: *Proceedings of the 2012 ACM International Conference on Supporting Group Work, GROUP ’12*, Association for Computing Machinery, New York, NY, USA, 2012, p. 273–276. doi:[10.1145/2389176.2389217](https://doi.org/10.1145/2389176.2389217).
- [5] D. Shahaf, C. Guestrin, Connecting the dots between news articles, in: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’10*, Association for Computing Machinery, New York, NY, USA, 2010, p. 623–632. doi:[10.1145/1835804.1835884](https://doi.org/10.1145/1835804.1835884).
- [6] D. Shahaf, C. Guestrin, Connecting two (or less) dots: Discovering structure in news articles, *ACM Trans. Knowl. Discov. Data* 5 (2012). doi:[10.1145/2086737.2086744](https://doi.org/10.1145/2086737.2086744).
- [7] B. F. Keith Norambuena, T. Mitra, Narrative maps: An algorithmic approach to represent and extract information narratives, *Proc. ACM Hum.-Comput. Interact.* 4 (2021). doi:[10.1145/3432927](https://doi.org/10.1145/3432927).
- [8] R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Litvak, The 6th international workshop on narrative extraction from texts: Text2story 2023, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), *Advances in Information Retrieval, Springer Nature Switzerland, Cham*, 2023, pp. 377–383.
- [9] J. Li, S. Li, Evolutionary hierarchical Dirichlet process for timeline summarization, in: H. Schuetze, P. Fung, M. Poesio (Eds.), *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 556–560.
- [10] F. ren Lin, C.-H. Liang, Storyline-based summarization for news topic retrospection, *Decision Support Systems* 45 (2008) 473–490. doi:<https://doi.org/10.1016/j.dss.2007.06.009>, special Issue Clusters.
- [11] S. Xu, S. Wang, Y. Zhang, Summarizing complex events: a cross-modal solution of storylines extraction and reconstruction, in: D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, S. Bethard (Eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 1281–1291.
- [12] P. Laban, M. Hearst, newsLens: building and visualizing long-ranging news stories, in: T. Caselli, B. Miller, M. van Erp, P. Vossen, M. Palmer, E. Hovy, T. Mitamura, D. Caswell (Eds.), *Proceedings of the Events and Stories in the News Workshop*, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1–9. doi:[10.18653/v1/W17-2701](https://doi.org/10.18653/v1/W17-2701).
- [13] D. Shahaf, C. Guestrin, E. Horvitz, Trains of thought: generating information maps, in: *Proceedings of the 21st International Conference on World Wide Web, WWW ’12*, Association for Computing Machinery, New York, NY, USA, 2012, p. 899–908. doi:[10.1145/2187836.2187957](https://doi.org/10.1145/2187836.2187957).

- [14] D. Shahaf, C. Guestrin, E. Horvitz, Metro maps of science, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 1122–1130. doi:10.1145/2339530.2339706.
- [15] M. Grootendorst, Bertopic: Neural topic modeling with a class-based tf-idf procedure, 2022. arXiv:2203.05794.
- [16] A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy, J. Heidecke, P. Shyam, B. Power, T. E. Nekoul, G. Sastry, G. Krueger, D. Schnurr, F. P. Such, K. Hsu, M. Thompson, T. Khan, T. Sherbakov, J. Jang, P. Welinder, L. Weng, Text and code embeddings by contrastive pre-training, 2022. arXiv:2201.10005.
- [17] E. Kamalloo, X. Zhang, O. Ogundepo, N. Thakur, D. Alfonso-hermelo, M. Rezagholizadeh, J. Lin, Evaluating embedding APIs for information retrieval, 2023. doi:10.18653/v1/2023.acl-industry.50.
- [18] OpenAI, New embedding models and API updates, 2024. URL: <https://openai.com/index/new-embedding-models-and-api-updates/>.
- [19] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.
- [20] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, MTEB: Massive text embedding benchmark, in: A. Vlachos, I. Augenstein (Eds.), Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 2014–2037. doi:10.18653/v1/2023.eacl-main.148.
- [21] D. M. Mistry, A. A. Minai, A comparative study of sentence embedding models for assessing semantic variation, in: Artificial Neural Networks and Machine Learning – ICANN 2023: 32nd International Conference on Artificial Neural Networks, Heraklion, Crete, Greece, September 26–29, 2023, Proceedings, Part X, Springer-Verlag, Berlin, Heidelberg, 2023, p. 1–12. doi:10.1007/978-3-031-44204-9_1.
- [22] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2020. arXiv:1802.03426.
- [23] L. McInnes, J. Healy, S. Astels, hdbscan: Hierarchical density based clustering, Journal of Open Source Software 2 (2017) 205. doi:10.21105/joss.00205.
- [24] L. van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (2008) 2579–2605.
- [25] A. Ghosh, M. Nashaat, J. Miller, S. Quader, Context-based evaluation of dimensionality reduction algorithms—experiments and statistical significance analysis, ACM Trans. Knowl. Discov. Data 15 (2021). doi:10.1145/3428077.
- [26] M. Sánchez-Rico, N. Hoertel, J. Alvarado, Combination of cluster analysis with dimensionality reduction techniques for pattern recognition studies in healthcare data: Comparing pca, t-sne and umap (2023). doi:10.31234/osf.io/zzvfv2.
- [27] M. Allaoui, M. L. Kherfi, A. Cheriet, Considerably Improving Clustering Algorithms Using UMAP Dimensionality Reduction Technique: A Comparative Study, Image and Signal Processing 12119 (2020) 317–325. doi:10.1007/978-3-030-51935-3_34.
- [28] S. J. Hwang, S. B. Damelin, A. O. H. III, Shortest path through random points, The Annals of Applied Probability 26 (2016) 2791 – 2823. doi:10.1214/15-AAP1162.
- [29] D. Shahaf, J. Yang, C. Suen, J. Jacobs, H. Wang, J. Leskovec, Information cartography: creating zoomable, large-scale maps of information, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, Association for Computing Machinery, New York, NY, USA, 2013, p. 1097–1105. doi:10.1145/2487575.2487690.
- [30] P. Zhou, B. Wu, Z. Cao, Emmbtt: A novel event evolution model based on tfidf and tdc in tracking news streams, in: 2017 IEEE Second International Conference on Data Science in Cyberspace

- (DSC), Shenzhen, China, 2017, pp. 102–107. doi:10.1109/DSC.2017.53.
- [31] J. Kleinberg, E. Tardos, *Algorithm Design*, Pearson, 2005.
 - [32] M. Pollack, Letter to the Editor—The Maximum Capacity Through a Network, *Operations Research* 8 (1960) 733–736. doi:10.1287/opre.8.5.733, publisher: INFORMS.
 - [33] A. P. Punnen, A linear time algorithm for the maximum capacity path problem, *European Journal of Operational Research* 53 (1991) 402–404. doi:10.1016/0377-2217(91)90073-5.
 - [34] V. Kaibel, M. Peinhardt, On the Bottleneck Shortest Path Problem, Technical Report 06-22, ZIB, Takustr. 7, 14195 Berlin, 2006.
 - [35] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik* 1 (1959) 269–271. doi:10.1007/BF01386390.
 - [36] M. Müller, Dynamic time warping, *Information retrieval for music and motion* (2007) 69–84.
 - [37] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, E. Keogh, Searching and mining trillions of time series subsequences under dynamic time warping, in: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*, Association for Computing Machinery, New York, NY, USA, 2012, p. 262–270. doi:10.1145/2339530.2339576.
 - [38] K. Wang, T. Gasser, Alignment of curves by dynamic time warping, *The Annals of Statistics* 25 (1997) 1251–1276.
 - [39] R. West, J. Pineau, D. Precup, Wikispeedia: an online game for inferring semantic distances between concepts, in: *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009, p. 1598–1603. URL: <https://dl.acm.org/doi/10.5555/1661445.1661702>.
 - [40] R. West, J. Leskovec, Human wayfinding in information networks, in: *Proceedings of the 21st International Conference on World Wide Web, WWW '12*, Association for Computing Machinery, New York, NY, USA, 2012, p. 619–628. doi:10.1145/2187836.2187920.
 - [41] B. F. Keith Norambuena, T. Mitra, C. North, Mixed multi-model semantic interaction for graph-based narrative visualizations, in: *Proceedings of the 28th International Conference on Intelligent User Interfaces, IUI '23*, Association for Computing Machinery, New York, NY, USA, 2023, p. 866–888. doi:10.1145/3581641.3584076.
 - [42] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, J. Stasko, vispubdata.org: A metadata collection about IEEE visualization (VIS) publications, *IEEE Transactions on Visualization and Computer Graphics* 23 (2017) 2199–2206. doi:10.1109/TVCG.2016.2615308.
 - [43] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: extraction and mining of academic social networks, in: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08*, Association for Computing Machinery, New York, NY, USA, 2008, p. 990–998. doi:10.1145/1401890.1402008.
 - [44] L. C. Freeman, Centrality in social networks conceptual clarification, *Social Networks* 1 (1978) 215–239. doi:[https://doi.org/10.1016/0378-8733\(78\)90021-7](https://doi.org/10.1016/0378-8733(78)90021-7).