

# 階層的ナレッジグラフを用いた 事件事例文書の述語中心の構造化手法

林瑛勲 森辰則  
横浜国立大学大学院

lim-younghun-sw@ynu.jp tmori@ynu.ac.jp

## 概要

本研究では、事件事例文書に含まれる事象を階層的ナレッジグラフ (KG) として構造化する手法を提案する。まず、KG のレイヤ構造を定義し、失敗知識データベース [1] を対象に自動取得システムを実装した。その結果、生成された KG の有用性を確認するとともに、応用可能性を検討した。開発したシステムは GitHub にて公開している<sup>1)</sup>。

## 1 はじめに

企業では、技術的なトラブルや装置の故障に関する問い合わせが日々発生しており、これらの対応には担当者が手作業で処理することが多い。このため、人的コストや対応時間が増大するという課題がある。近年、大規模言語モデルをはじめとする AI 技術の発展により、自動応答システムが導入され、簡単な問い合わせに対応できるようになった。しかし、技術トラブルや装置故障のような複雑な問題については、AI では十分に対応できず、依然として人間に依存している。本研究は、これら複雑な問題に対応可能な自動問い合わせ応答システムの実現を目指し、その基盤となる事件事例文書の構造化技術を提案する。

事故に関する情報は通常、事件事例文書として記録される。この文書には、特定の場所や時間で発生した小さな事象 (イベント) が相互に影響を及ぼし、最終的に原因事象から結果事象へ至るプロセスが含まれている。しかし、これらの情報は文書内に散在しており、人間や AI が十分に理解するには多大な労力を要する。そこで本研究では、事象間の相互作用や因果関係を効果的に整理・可視化するため、グラフ構造を採用したナレッジグラフ (Knowledge Graph:KG) を構築することにした。

さらに、本研究では KG に階層構造を導入し、文書が持つ分類や章立てなどの階層的な情報も保持できるように設計した。この階層構造により、水平方向では事象間の関係性を、垂直方向では文書のツリー構造を可視化できるという利点がある。本研究では、こうした階層的 KG の構築方法を定義し、GPT などの大規模言語モデルを用いた自動生成プロセスを提示する。また、失敗知識データベース [1] を用いてその有効性を確認し、応用可能性について議論する。

## 2 関連研究

### 2.1 事件事例文書の構造化

先行研究として、東明らの研究 [2] では事件事例文書の「事象概要」部分を対象に、「状態」「状態変化」「思考・判断」の三種類を設定して事象を構造化する手法を提案している。しかし、「事象概要」は文書全体を簡潔にまとめたものであるため、同一事象が異なる表現で複数箇所に散在するケースは想定されておらず、対処できていない。また、三種類のみ分類では文書に含まれる多様な記述を十分に網羅しきれないという限界がある。

そこで本研究では、こうした課題を解決するため、文中の多様な記述を述語とエンティティの観点で整理し、一つの事象を一つの述語またはエンティティとして対応付ける手法を導入する。これにより、従来のように「事象概要」などの要約部のみを扱うのではなく、原因・経過・対策など文書内の複数セクションに散在する情報も含めて全体を包括的に構造化でき、同一の事象が異なる章や節で繰り返し言及される場合でも一貫した表現が可能となる。また、事故情報をもつ潜在的な多様性を捉えやすくなるため、将来的な大規模データへの展開や複雑な事例分析にも適用しやすいと考えられる。

1) [https://github.com/four4minutes/NLP2025\\_HEKG](https://github.com/four4minutes/NLP2025_HEKG)

## 2.2 ナレッジグラフとその応用

Knowledge Graph(KG)は、2012年にGoogleが提唱して以来、世界中の知識をグラフ構造で表現する試みが多く行われてきた[3]。しかし、静的な情報が中心のKGでは、時間経過や事象の流れを十分に表現できないという課題がある。そこで、イベント(事象)に焦点を当てた動的な手法としてEvent Knowledge Graph(EKG)[4]などが提案されてきた。EKGは事象中心の情報を豊かに扱える一方、実運用に向けた精度面などに課題が残っている。本研究は、EKGをベースにした発展的アプローチの一つであり、後続研究の位置づけをもつ。

また、Event Cubel[5]は複数の概念次元とイベントをそれぞれ独立した階層構造で捉える点が特徴であるが、階層同士が分離しているため直観的な把握はやや難しい。本研究では、文書内の各階層を相互に連結し、全体を一貫した構造として統合するアプローチをとることで、事故事例文書に特化した可視化と活用を可能にしている。

さらに、本研究はAIモデルとの連携を視野に入れる点でGraph RAG[6]に類似するが、Graph RAGはQuery Focused Summaryを主目的としており、グラフ構造を厳密に定義していない。一方、本研究では事故事例文書を階層的に整理する手法自体を重視しており、Graph RAGとはアプローチが異なる。

## 3 ナレッジグラフの構造の定義

### 3.1 グラフ構造の概要

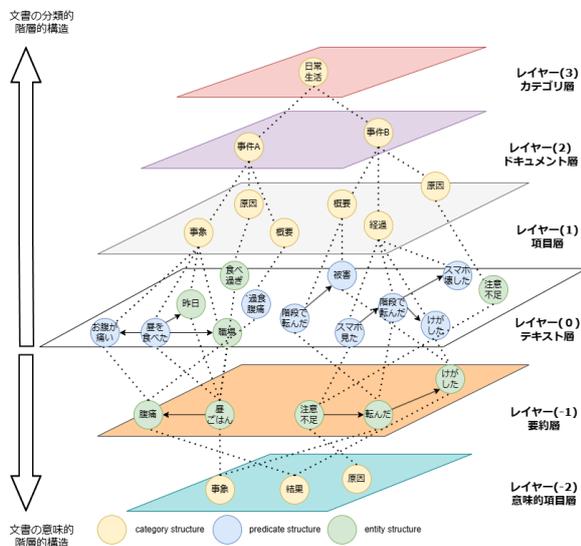


図1 グラフ構造の概念図

本研究で提案するKGは、事故事例文書の情報を合計6つのレイヤに分割して階層的に表現するように設計している(図1)。具体的には、レイヤ(0)を基点として、文書集合の索引構造を上方向に、文書の内容から抽出した要約・意味情報を下方向に配置し、上下へ拡張する形をとる。以下では、それぞれのレイヤと階層間の関係を簡潔に述べる。

- **カテゴリ層 (レイヤ(3))**

カテゴリごとに整理されている文書集合を表す層であり、「機械関連」「化学関連」などのカテゴリ情報をノードとして持つ。文書集合を俯瞰する最上位概念に相当し、下位の文書を統括する役割を担う。

- **ドキュメント層 (レイヤ(2))**

カテゴリに紐づく具体的な文書を表す層である。たとえば「東京ビッグサイトエスカレータ逆走(2008年)」「エキスポランドジェットコースター事件(2007年)」などが該当する。

- **項目層 (レイヤ(1))**

各ドキュメントが内部で持つセクションを表す層で、「事象概要」「原因」「経過」といった事故報告文書特有の項目をノードとして整理する。文書構成を大まかに把握する基盤となる。

- **テキスト層 (レイヤ(0))**

項目層に含まれる実際のテキストデータをKG上で表現する層である。たとえば「原因」項目に書かれた「左側の車輪を支える車軸ブロックの軸が折断した。」といった文からは、4章で述べる処理を通してノードが生成され、その関係性をエッジで結び付けられる。

- **要約層 (レイヤ(-1))**

テキスト層で抽出した事象情報を、同一または類似内容を統合した層である。文書内の複数箇所に散在する重複事象をまとめることで、大規模データにおける冗長性を低減し、より一貫した参照が可能になる。

- **意味的項目層 (レイヤ(-2))**

要約層で統合されたノードに対し、さらに意味的な役割や概念的情報を付与する層である。

図1を俯瞰すると、テキスト層(レイヤ(0))を基点に、上方向(項目層→ドキュメント層→カテゴリ層)へ進むにつれて文書全体を分類する範囲が広がり、下方向(要約層→意味的項目層)へ進むほど記述が統合・抽象化され、高次の意味情報を扱えるよ

うになる。このように二方向の階層を組み合わせることで、文書の分類索引を把握すると同時に、事象の要約や抽象化も効率よく行え、事象事例文書を多面的に整理することが可能となる。

## 3.2 ノードとエッジの定義

### 3.2.1 述語項構造とエンティティ

事象事例文書には、事故の流れや原因、対処方法などが記述されており、複数の小さな事象が時間とともに互いに作用しながら進行している。本研究では、こうした因果関係を的確に捉えるため、文章中の述語と格要素を合わせて扱う述語項構造を導入し、一つの述語項構造を一つの単位事象としてみなす。たとえば「列車の支持軸が破裂して列車が脱線した」という文では、「破裂」と「脱線」という述語を手がかりに、複数の事象が連鎖して大きな事故に至る流れを説明できる。

さらに、事故の進行以外にも、「AはBである」のような概念的な記述が含まれ、AとBが同一概念または属性関係にある場合がある。このような情報は述語項構造だけでは十分に表現できないため、エンティティを別のノードとして導入し、事象（述語項構造）と組み合わせることで文書内の多様な情報をよりの確に捉えられるようにしている。

### 3.2.2 ノード

本研究のKGでは、文書を構造化するために大きく3種類のノードを定義しており、それぞれ内部に目的に応じた情報を保持する。以下に各ノードの役割とその例を図2から図4に示す。

**category structure** : 文書を大区分で分類するノードであり、文書や項目などカテゴリ情報を表す(図2)。

category_structure	例
index : インデックス番号	index : 50
hierarchical_level : 階層構造の情報	hierarchical_level : 1
category_type : カテゴリの種類	category_type : 項目名
category_title : カテゴリの内容	category_title : 原因

図2 category structure ノード

**predicate structure** : 事象の流れといった動作・状態を、述語項構造で表したノードである(図3)。

predicate_structure	例
index : インデックス番号	index : 48
hierarchical_level : 階層構造の情報	hierarchical_level : 0
agent_argument : 述語項構造における格	agent_argument : ジェットコースターの車輪(格)
predicate : 述語項構造における述語	predicate : 脱線(述語)
argument : 述語項構造における格以外の各要素	argument : レール(カ格)
modifier : 適用修飾語	modifier : 突然(修飾)

図3 predicate structure ノード

**entity structure** : 述語項構造の表現が難しい記述や概念的な要素を、エンティティとして表すノードである(図4)。

entity_structure	例
index : インデックス番号	index : 25
hierarchical_level : 階層構造の情報	hierarchical_level : 0
entity : 内容	entity : ゴールデンウィークの午後

図4 entity structure ノード

### 3.2.3 エッジ

本研究のKGでは、ノード間の事象に関連する関係を表1に示すように複数種類のラベル付きエッジとして定義している。たとえば(x) - [sub] → (y)は、「sub」というラベルが付与されたエッジであり、ノードyがノードxの分類の階層構造における下位に位置することを示す。一方、ノード同士の概念的な関係については事前定義を行わず、GPT-4oを用いて文脈に応じたエッジラベルを自動生成する。

形式	意味
(x) - [sub] → (y)	(y)は(x)の分類の階層構造において下位
(x) - [correspond_to] → (y)	(y)は(x)の意味的階層構造において下位
(x) - [SpecificTime] → (y)	(y)は(x)に詳細な時間情報を提供する
(x) - [SpecificPlace] → (y)	(y)は(x)に詳細な場所情報を提供する
(x) - [next_TimeStamp] → (y)	(y)のTimeStampは(x)の直後
(x) - [explain_details] → (y)	(y)は(x)に詳細な情報を提供する
(x) - [explain_reason] → (y)	(y)は(x)に対する理由を説明する
(x) - [explain_cause] → (y)	(y)は(x)に対する直接的な原因である
(x) - [equivalent] → (y)	(y)と(x)は類似した情報である

表1 事象に関連するエッジの種類

## 4 ナレッジグラフの自動取得

第3章で定義したグラフモデルに基づき、失敗知識データベースの文書に対してナレッジグラフを自動的に生成するシステムを実装した。これにより、本研究で提案した手法を自動化の観点から検証できる。本システムは失敗知識データベースに含まれる全文書を入力とし、最終的に図5のような形でグラフ構造を提示する。入力データは上から順にパイプラインで処理され、1から5までのステップを経てKGが生成される。一部の工程ではGPT-4oモデルを活用し、特定の文書「東京ビッグサイトエスカレータ逆走」をfew-shots学習の例として利用した。

1. **前処理** : 失敗知識データベースの文書をJSON形式に変換する。

2. **ノードの生成**

(1) 文からカテゴリに該当する記述を抽出し、category structure ノードを生成する。

(2) 文から時間表現と場所表現を抽出し、entity structure ノードを生成する。テキスト解析にはGPT-4oを使用する。

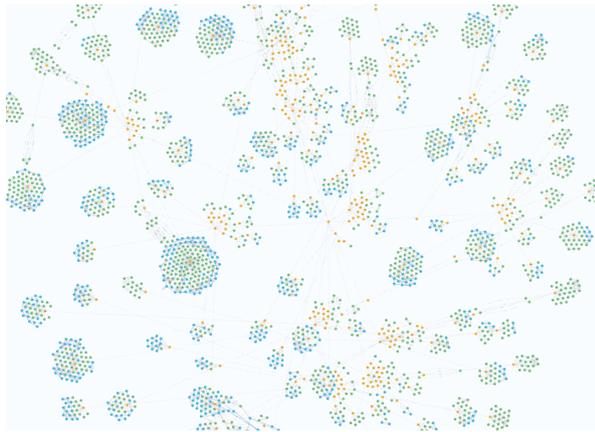


図5 ナレッジグラフの全体像

(3) GPT-4o を用いて文から述語を抽出し、事象に関する述語と概念に関する述語を分類する。さらに、抽出された述語を再度 GPT-4o に提示し、述語項構造を導出するよう指示し、その結果を predicate structure ノードとして生成する。概念に関する記述については、該当述語を参照して entity structure ノードを生成する。述語の特定および述語項構造の抽出には、京都大学コーパス [7] と NAIST コーパス [8] で提示されたタグ付け基準を GPT-4o のプロンプトに提示して実装した。

### 3. エッジの生成

(1) 文書の分類的階層構造をもつノード間に sub 関係を付与する。また、意味的階層構造をもつノード間には correspond\_to 関係を付与する。

(2) 時間表現と場所表現のノードについて、それが表す事象ノードと SpecificTime 関係および SpecificPlace 関係を付与する。

(3) Event Evolution[9] で定義されたスコアに基づき、 $CS \cdot \sqrt{TP \cdot DF}$  と Static Thresholding pruning ( $\lambda = 0.60$ ) を用いて事象ノードの時間順を調整し、ノード間に next\_TimeStamp 関係を付与する。最後に GPT-4o を用いて結果を検証し、誤りが見つかった場合のみ手動で修正する。

(4) GPT-4o を活用して手がかり表現 [10] に基づき、原因事象と結果事象の因果関係を抽出し、explain\_cause または explain\_reason 関係を付与する。

(5) あるノードに関する詳細情報を提供するノードには explain\_details 関係を付与する。

(6) 類似度計算を行い、一定の閾値を超えるノード同士には equivalent 関係を付与する。類似度判定には sentence-transformer の stsb-xlm-r-multilingual[11] を用い、同一または類似する情報を統合する。

(7) GPT-4o を用いて概念を表す関係を生成・付与

する。このとき、関係が乱立しないよう辞書を活用して既存の関係を確認したうえで、新たに必要な関係のみ追加する。

4. グラフ DB への埋め込み: 分析結果を CSV ファイルに出力し、Neo4j Community Edition<sup>2)</sup>を用いてグラフ DB を構築する。

### 5. グラフに基づく再構築

(1) 類似したノード (equivalent 関係、explain\_details 関係、SpecificTime 関係、SpecificPlace 関係を持つノードの塊) を 1 個のノードに統合することで冗長表現をまとめ、情報を凝縮する。

(2) 統合されたノードを GPT-4o モデルで分析し、文章を整理する。その結果を基に、レイヤ (-1) 要約層にノードを生成する。

(3) 文の意味役割を GPT-4o モデルで判定し、レイヤ (-2) 意味的項目層にノードを生成する。

## 5 ナレッジグラフの応用例

本研究で構築したナレッジグラフ (KG) は、事故事例文書の事象や原因、対策を階層的に整理するため、高度な検索・因果関係推論、Q&A など多彩な応用が期待できる。また、文書間に散在する同一または類似事象を統合することで、事故の再発防止策や原因追跡を効率よく行う基盤としても有用である。

さらに、ナレッジグラフを活用した要約文の生成も大きな利点となる。事故事例文書は記述が膨大かつ複雑で、生成型 AI に単純に「要約してほしい」と依頼すると誤った情報 (いわゆるハルシネーション) が混入しやすい。本研究では、KG から得られる正規化された (重複表現などを除去して一貫性を持たせた) 情報を基に要約を行うことで誤情報を抑えながら原因と結果、進行順序を明確に示せることを確認しており、GPT-4o による単純要約との比較例は付録に掲載している。

## 6 おわりに

本研究では、事故事例文書の複雑な記述を階層的に整理するための KG 構築手法を提案した。実際の文書データを用いた自動生成システムにより、本手法が実装可能であることを確認し、要約文生成タスクへの応用例も示した。一方、自動取得手法のさらなる改善や他分野への適用は今後の課題である。本手法は、事故事例文書の有効活用寄予する基盤となると期待される。

2) <https://github.com/neo4j/neo4j>

## 参考文献

- [1] 失敗知識データベース, (2025-1 閲覧). <http://www.shippai.org/fkd/index.php>.
- [2] 東明幸太, 福岡康大, 森辰則, 伊藤拓海. 事件事例構造化コーパスの構築. 言語処理学会第 29 回年次大会発表論文集, p. 1074–1078, 2023.
- [3] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D’amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, Antoine Zimmermann. Knowledge graphs. **ACM Comput. Surv.**, Vol. 54, No. 4, July 2021.
- [4] Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. What is event knowledge graph: A survey. **IEEE Trans. on Knowl. and Data Eng.**, Vol. 35, No. 7, p. 7569–7589, July 2023.
- [5] Qing Li, Yun Ma, and Zhenguo Yang. Event cube – a conceptual framework for event modeling and analysis. In Athman Bouguettaya, Yunjun Gao, Andrey Klimenko, Lu Chen, Xiangliang Zhang, Fedor Dzerzhinskiy, Weijia Jia, Stanislav V. Klimenko, and Qing Li, editors, **Web Information Systems Engineering – WISE 2017**, pp. 499–515, Cham, 2017. Springer International Publishing.
- [6] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. **arXiv preprint arXiv:2404.16130**, 2024.
- [7] 河原大輔, 黒橋禎夫, 橋田浩一. 「関係」タグ付きコーパスの作成. 言語処理学会第 8 回年次大会発表論文集, pp. 495–498, 2002.
- [8] 飯田龍, 小町守, 井之上直也, 乾健太郎, 松本裕治. 述語項構造と照応関係のアノテーション: naist テキストコーパス構築の経験から. 自然言語処理, Vol. 17, No. 2, pp. 2.25–2.50, 2010.
- [9] Christopher C Yang, Xiaodong Shi, and Chih-Ping Wei. Discovering event evolution graphs from news corpora. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, Vol. 39, No. 4, pp. 850–863, 2009.
- [10] Hiroki Sakaji, Satoshi Sekine, and Shigeru Masuyama. Extracting causal knowledge using clue phrases and syntactic patterns. In Takahira Yamaguchi, editor, **Practical Aspects of Knowledge Management**, pp. 111–122, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [11] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 11 2019.

## A 要約文の結果

失敗知識データベースに記載された「エキスポランドジェットコースター事故（2007年）」の一部を対象に要約を生成し、その結果を以下に示す。生成された要約文に含まれる誤情報には下線を付して示した。

GPT-4oに「与えられた文から事象を時間順に整理し、事象の原因も含めて説明する要約を作成してください」と指示し、要約文を生成した。一方、本モデルで生成された要約文は、要約層と意味的項目層からのノードとエッジの情報をGPT-4oに与え、表現の言い換えを許可しない条件下で依頼した。

### A.1 GPT-4oによる要約文

- 事象を時間順に整理

2007年1月30日、千里万博公園内の遊園地「エキスポランド」にて、立ち乗りジェットコースター「風神雷神II」の定期検査が実施された。検査は目視で行われ、すべての検査項目は問題なし（「A」評価）として報告された。

2007年5月5日のゴールデンウィーク中に、「風神雷神II」と「雷神」がペアで交互に運行されていた。午後12時50分頃、「風神雷神II」が22名を乗せてレール上を走行中、終点手前の約300m付近で突然事故が発生した。ジェットコースターの2両目が突然レールから脱輪し、車体が大きく左側に傾き、左側の鉄柵に衝突した。これにより、乗客1名が頭部を強打して死亡し、他の乗客21名が負傷した。また、事故を目撃した13名の入場客がショックで体調を崩し、病院へ搬送された。

- 車両脱輪の原因

2両目の車両の左側の車輪を支える軸が疲労破壊により折損し、車両が脱輪。疲労破壊の原因は、軸を固定するナットの締結不十分により、軸部分で繰り返し応力（疲労）が蓄積したためである。最終的に疲労破壊が起きた軸は、破断面の70%以上が既に亀裂で覆われており、繰り返しの応力により徐々に弱化していた。

### A.2 本モデルによる要約文

- 事象

2007年1月30日、ジェットコースターの定期検査が目視により実施された。そして、すべて

の検査項目が「A（指摘なし又は良好）」と報告されていた。

2007年5月5日、ゴールデンウィーク中に、立ち乗りジェットコースター「風神雷神II」が6両編成で運行していた。その際、ジェットコースター「風神雷神II」に乗客22名が乗車し、レール上を走行していた。ところが、両目左側の車輪を支える合金製の軸が突然折損し、ねじ部が切断した。その後、ジェットコースターの2両目の車輪がレールから突然脱輪した。そして、車体が左側に大きく約45度傾いた。その結果、搭乗者1名が車体と鉄柵に頭を挟まれ、死亡した。さらに、負傷者21名が出た。加えて、入場客13名が事故を目撃して気分を悪くし、病院へ搬送された。

- 原因

その原因はねじ部の疲労破壊である。そして、ナットが軸を固定する締結が不十分だった。また、繰り返し応力が低いことも原因であった。最終破断面が全破面の70%以上を占めていた。