計算工学講演会論文集　Vol.30 (2025年6月)　日本計算工学会

# CAEデータベースにGraphRAGを適用する方法（"計算事例","質疑応答","入力データ"等が,Directory階層構造を構成している場合）

Application of GraphRAG to a CAE database
("Calculated cases", "Questions and answers", "Input data", etc.,
are in the Directory hierarchical structure)

吉田史郎[1)]
Shiro Yoshida

1) 湘南技術開発株式会社（〒251-0035 神奈川県藤沢市片瀬海岸3-19-17 湘南ホームズ402号

E-mail: Sir_Yoshida@jnifty.com

In the case that "calculation cases", "questions and answers", "input data", etc. constitute a directory hierarchical structure,
(1) Treat each category (case studies, questions and answers, input data) as a node.
(2) The logical dependencies among categories are defined as edges.
(3) Using the nodes and edges, we construct and visualize the overall GraphRAG.
As a result, we report a case in which it is possible to quickly find a computation case based on specific input data. We report a case study in which it became possible to quickly find a computation case based on specific input data.

***Key Words :*** *Clustering, k-means method, Langchain, GraphRag, Chunking, Regression curve*

## 1.　はじめに

厳密に構造化されていない自然言語のデータベースを探索する手法としてRAG(Retrieval Augmented Generation)と呼ばれる手法が注目されている．これはデータベースを構成する自然言語ファイルの中から,代表的なキーワードを拠り所にして,目標とするファイルを探し出す手法であるが,

(1) 各ファイルに含まれる単語を,一定の規則でベクトル化する.

(2) 各ファイルに含まれる「文章」を短い「文章の断片」に分割する.

(3) ベクトル化された数値データと「文章の断片」を,各種クラスタリング手法で,「類似度」を検証する.

(4) その結果,ベクトル化された数値データの比較だけで検証する際の信頼性を向上させる事が期待される[1].

(5) しかし,CAEデータファイルのように,定型的な単語が多く含まれる場合,どのような工夫が求められるか,検討する必要がある.

(6) そこで,典型的な数値計算の入出力関係を調べて,その適用可能性を検討する.

## 2.　GraphRAGの活用

自然言語データを短い「文章の断片」に分割して,クラスリングする操作を「Chunk分割」と呼び,例えば文ごと,段落ごと,意味的にまとまった単位ごとに分割する.

ベクトル化 & クラスタリング

(1) 分割された文字列（チャンク）をベクトル化
一般的には Sentence-BERT (SBERT)[2], OpenAI Embeddings[3], TF-IDF[4], Word2Vec[5] 等を利用して,各チャンクを数値ベクトルに変換し,高次元空間上の分布を取得する.

(2) クラスタリング
K-Means[6], DBSCAN[7], Agglomerative Clustering[8], HDBSCAN[9]
等を適用して,文の意味的な類似度に基づいてクラスタリングする.

(3) Node2Vec[10] の活用
・Node2Vec はグラフデータを埋め込みベクトルに変換する手法で,自然言語データをグラフ構造で表現する.
・そして文や段落をノードとして扱い,意味的な類似性が検出されたノードの間を,エッジと呼ばれる線分で接続する.
・更に,Sentence-BERT ,
等の機能を活用して,類似度スコアを計算し,エッジの重みを設定する.
・次に,Node2Vec でノード（チャンク）の埋め込みを取得することにより,ベクトル化とクラスタリングの前処理として活用する.
・ここまでの手順は,GraphSAGE[11] や DeepWalk[12] などの手法と組み合わせることも可能である.

(4) GraphRAG Toolkit[13] の活用
GraphRAG は,(1), (2) を組み合わせることで,より高精度な情報検索やクラスタリングを実現できる.
・GraphRAG による精度向上のポイントは,クラスタリン

グだけでなく,グラフ検索 (Graph-based Retrieval) にも活用できる.
・Node2Vec で取得したベクトルを用いて,関連するチャンクを検索すると,既存の RAG (Retrieval-Augmented Generation)
モデルと統合することで,意味的に近いテキストをより正確に検索・利用できる.
(5) 分割された文字列そのものは, Node2Vec でベクトル化することで,有機的な機能を果たす.
即ち,GraphRAG Toolkit を活用することで,ベクトル検索とグラフ検索のハイブリッドで高精度な情報取得が可能になる.
・GraphRAG が適する場面
GraphRAG は本来,自然言語処理 (NLP) において,情報検索 や 文の意味的関係を考慮した処理を行うために設計されている.
特に,以下のようなケースで有効である：
a)テキストの意味がグラフ構造として表現できる場合：
例: 法律文書,ニュース記事,FAQ などの文脈情報を活用して検索する.
b)複雑な関係性を持つデータを検索して推論する場合：
例: 知識グラフ (Knowledge Graph),RAG モデルでの情報検索
c)文書間の関連性を動的に検索し,最適なコンテンツを提供する場合：
例: Q&A システム,企業内のドキュメント検索
d)点のつながりに「意味」を持たせたい場合：
例：時系列データや,クラスター間のつながりが重要な場合
e)点同士の「関係性」や「遷移」を学習したい場合：
Node2Vec でグラフ埋め込みを行い,点の関係性を数値化する.
f)クラスタリングだけでなく,「クラスタ間の関係」も分析したい場合：
例：点のグループ間のつながりを分析し,より精度の高いクラスタリングを実現できる.

## 3. GraphRAGが不要な場合（機能過剰）

単純な点のグループ化（クラスタリング）をしたいだけのケース
a)K-Means, DBSCAN, Agglomerative Clustering などのクラスタリング手法で十分点同士のつながりが明確でない場合：
b)GraphRAG を使っても,点同士のエッジが適切に構成できないデータが単純な幾何学的な分布を持つ場合：
c)楕円の点が既知のパターン（例: 数式で表現可能）ならば,グラフ構造の適用は不要
・基本的な点クラスタリングには,GraphRAG は不要（機能過剰）である.
・ただし,点の関係性を学習し,より高度な分析をする場合は適用する価値がある.
・「単純なグループ分け」ではなく,「点のつながり」や「動的な構造」を考慮したい場合, GraphRAG を活用することに,積極的な意味を見出せる.

## 4. 楕円曲線の復元

(1) 楕円重ね合わせ(点列データ)から,本来の楕円の数式を逆算する.

楕円の数式

一般的な2D楕円の方程式は以下のように示される：

$$\frac{(x-h)^2}{a^2}+\frac{(y-k)^2}{b^2}=1 \quad (1)$$

ただし,
(h, k)*h, k*：楕円の中心
a：長軸の半径
b：短軸の半径
3D空間の場合,回転行列を考慮すると,以下のように拡張される：
Ax2+Bxy+Cy2+Dx+Ey+F=0*Ax2+Bxy+Cy2+Dx+Ey+F=0* *(2)*

A,B,C,D,E,F を求めることで,楕円の形状を復元できる.

### (2) 最小二乗法を用いた楕円フィッティング

・点列データから 「非線形回帰」 を使ってパラメータ A,B,C,D,E,F を推定する.

・例えば,「楕円フィッティングアルゴリズム 」[14] を適用すると,最適な楕円を近似できる
・Python には numpy.linalg.lstsq() , scipy.optimize.curve_fit() など備わっており,有効な機能を果たす..

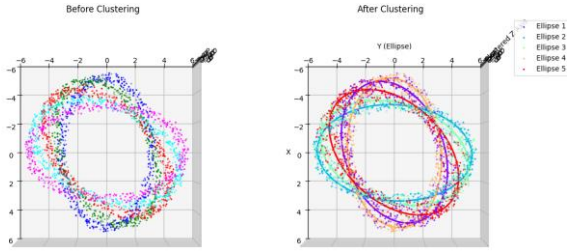


**Fig.1 Regressed Ellipse Curve**

## 5. 正弦曲線の復元

GraphRAG Toolkit は「ローカルミニマム回避」の直接的な解決策にはならない.しかし,
「ローカルミニマム」を避けるための知識選択・モデル設計支援」においては非常に役立つことが期待される.
(1) 推論や検索の補助ツール GraphRAG（Graph Retrieval-Augmented Generation）は,ノード（知識）エッジ（関連性）で構成される「知識グラフ + RAG型検索」を使って,文脈に応じて重要な情報を抽出・参照する.
(2) curve_fitのローカルミニマム問題
curve_fit
や類似の非線形最適化手法は,「初期値やノイズ」の影響に敏感である.したがって誤った局所最適値にたどり着きやすい.
即ち,複数の局所最小値が存在する場合,初期値設定が不適切だと,不正確な解に収束する.またノイズが多いと,過学習しやすい.
(3) GraphRAG が間接的に役立つ場面：
a)正しい関数モデルを見つけたい場合：
b) 類似事例・数式の文脈を検索 したい場合：
c)適切な初期値設定の知識が欲しい場合：
d) モデルごとの初期値や事前推定方法の検索する場合：

e) ノイズに強い回帰法の候補を探したい：
f) Robust回帰や周波数ドメイン変換などの提案を検索する場合：
(4) GraphRAG Toolkit を活用して,動的に“適切な初期値”を探索するcurve_fit のような非線形最小二乗法では,誤った周波数パラメータから始まると,ローカルミニマムに陥り,誤った振動数でフィットされる危険性がある.
GraphRAG は「グラフ型知識」と「RAG（検索）」を融合したものである.
その機能を活用することで,各クラスタの波形に応じた適切な“フィッティング初期値”を提案する仕組みを作ることができる.
ステップ 1：
a) 各クラスタ波形の特徴を抽出する：
b) 各クラスタについて,以下に示す項目を抽出する：
・特徴量
・使い方
・周波数スペクトル（FFT）
・高速フーリエ変換で主成分を抽出
・ピーク数 / ゼロ交差数
・周波数の見積もりに使える 信号長
ステップ 2：
c) GraphRAG Toolkit を以下のように活用する：
・各クラスタの波形特徴を「ノード」として登録
・ノード間の類似性（周波数帯域や波形形状）でエッジを構成
・対象クラスタに最も近い「過去の波形事例」から p0（初期値）を推薦
ステップ 3：
・GraphRAG に質問形式で初期値を問い合わせる
・GraphRAG の検索APIなどを使った事例：
・質問:
「入力波形がゼロ交差12回」
「振幅約1.0」
「ピーク間隔約0.1秒」
「最適な sin カーブの初期周波数と位相は？」 GraphRAG は過去のフィット事例から：
・python コピーする
・編集する
・回答: 「p0 = [1.0, 12π, 0.0] が妥当です.」
のような返答を生成できる.

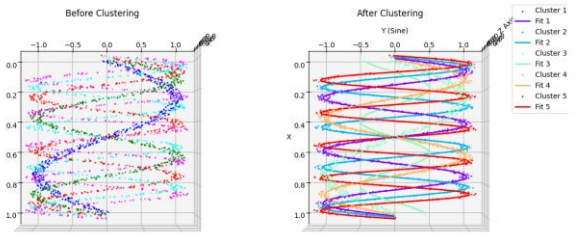


**Fig.2 Regressed Sinusoid Curve**

(1) CAE データベースの Directory 階層構造に RAG を適用する方法
(2) データの構造理解: :
・データベースの階層構造を詳細に分析し,主要なカテゴリ（計算事例,質疑応答, 入力データ）を識別する.
(3) グラフ構築の準備:
・ 各ディレクトリとサブディレクトリをノードとして考える.
・ ファイルやフォルダ間の関連をエッジとして表現するためのルールを定義する.
例：同じプロジェクトやトピックに基づくファイルは,強い接続（エッジ）を 持つと仮定する.
(4) 関連データの特定:
・ディレクトリ内のファイルからメタデータやキーワードを抽出する.
・それらがどのよ うに関連しているかを分析する.
例：「計算事例」内の特定の計算が「入力データ」に直接リンクしている場合,これらの間にエッジを設定する.

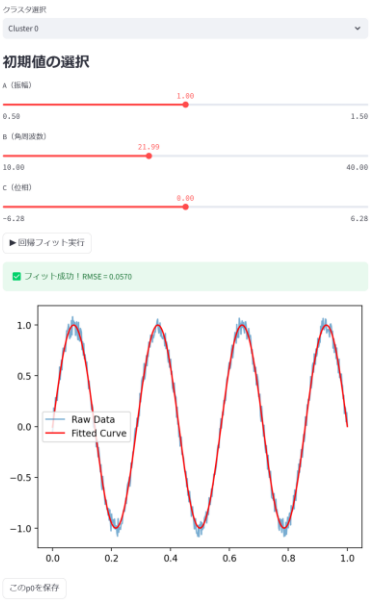


**Fig.3 Search for Initial Frequency**

## 6. 階層構造Directoryの取り扱い

「計算事例」,「質疑応答」,「入力データ」などが, Directory 階層構造にな っているCAEデータベースの事例[15]を Fig.4に示す.


**Fig.4 Hierarchically structured Directory**

(5) グラフの構築:
・抽出したデータと定義したルールに基づき,ノード間にエッジを設定する.
・その結果,全体としてのデータのつながりや流れを視覚的に表現するグラフが完成する.
(6) 検索機能の統合:
・RAGを活用して,ユーザーが必要とする情報に迅速かつ効率的にアクセスできるようする.

例：特定の「計算事例」を検索すると,関連する「質疑応答」や
「入力データ」へのリンクが提供されるように設定する.
(7) システムの最適化と更新:
・データベースが更新されるたびに,グラフも適宜更新して,最新の情報が反映さ れるようにする.
・このプロセスを通じて,複雑なデータベース内の情報がよりアクセスしやすく整理できる.
・その結果,利 用者が必要とする情報に素早くアクセスできるようになる.
・その際,データベースの規模や更新 の頻度に応じて,このプロセスの各ステップを調整することが重要である.

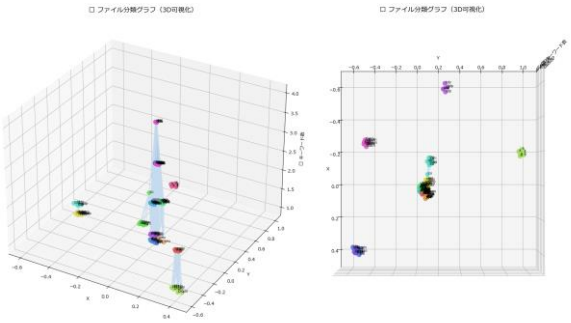

**Fig.5 Search for Initial Frequency**

## 7. おわりに

(1) 厳密に構造化されていない自然言語のデータベースを探索する手法としてRAG(Retrieval Augmented Generation) と呼ばれる手法が注目されている.
(2) これはデータベースを構成する自然言語ファイルの中から,代表的なキーワードを拠り所にして,目標とするファイルを探し出す手法である.
(3) しかし,CAEデータファイルのように,定型的な単語が多く含まれる場合,どのような工夫が求められるか,検討する必要がある.
(4) そこで,典型的な数値計算の入出力関係を調べて,その適用可能性を検討した.
(5) その結果,階層化されたCAEデータベースにおいては,有効なファイル探索手順を導き出すヒントが得られた.

### 参考文献

[1] 園田 憲一,日本オラクル株式会社, “グラフDBの基礎と生成AIアプリへの応用”, Available at
https://qiita.com/ksonoda/items/98a6607f31d0bbb237ef
[2] Zenn, “Sentence-BERTを用いた高速な類似文書検索”, Available at
https://zenn.dev/skwbc/articles/sentence_bert_implementation
[Accessed Feb 14 2025]
[3] OpenAI, “Introducing text and code embeddings”, Available at
https://openai.com/index/introducing-text-and-code-embeddings/
[Accessed Feb 14 2025]
[4] Zenn, “【自然言語処理】【Python】TF-IDFを使って文書の特徴をつかもう”, Available at
https://zenn.dev/robes/articles/241f6c3fac1486
[Accessed Feb 14 2025]
[5] Wikipedia, “Word2vec”, Available at
https://ja.wikipedia.org/wiki/Word2vec
[Accessed Feb 14 2025]
[6] Wikipedia, “k-means”, Available at
https://ja.wikipedia.org/wiki/K%E5%B9%B3%E5%9D%87%E6%B3%95
[Accessed Feb 14 2025]
[7] Wikipedia, “DBSCAN”, Available at
https://ja.wikipedia.org/wiki/DBSCAN
[Accessed Feb 14 2025]
[8] hdbscan, “Comparing Python Clustering Algorithms”, Available at
https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html
[Accessed Feb 14 2025]
[9] hdbscan, “Comparing Python Clustering Algorithms”, Available at
https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html
[Accessed Feb 14 2025]
[10] Deus Ex Machina, “Node2Vecの概要とアルゴリズム及び実装例について”, Available at
https://deus-ex-machina-ism.com/?p=56947
[Accessed Feb 14 2025]
[11] Deus Ex Machina, “GraphSAGEの概要とアルゴリズム及び実装例について”, Available at
https://deus-ex-machina-ism.com/?p=56943
[Accessed Feb 14 2025]
[12] Deus Ex Machina, “DeepWalkの概要とアルゴリズム及び実装例について”, Available at
https://deus-ex-machina-ism.com/?p=56945
[Accessed Feb 14 2025]
[13] Ian Robinson, Abdellah Ghasse, “Introducing the GraphRAG Toolkit”, Available at
https://aws.amazon.com/jp/blogs/news/introducing-the-graphrag-toolkit/, [Accessed Feb 14 2025]
[14] Fitzgibbon, Pilu, Fisher,(1996), “Direct Least Square Fitting of Ellipses”, Available at
https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/PILU1/demo.html
[Accessed Feb 14 2025]
[15] CAE大事典編集ワーキンググループ, “CAE実用大事典”, Available at
https://srm.nc-net.or.jp/dictionary/cae/
[Accessed Feb 14 2025]