第4章 テキストをベクトルとして表現するEmbedding（埋め込み）…数学的な処理で定量的な判断が可能に

実験2…正常データとの類似度を使った異常検知

氏森 充

表1　主なLLMのEmbedding対応状況

分　類	例	Embedding対応可否	備　考
Embedding対応APIを提供しているLLM	OpenAI（text-embedding-3-large），Voyage，Cohere	可能（公式サポート）	専用エンドポイントで高速に取得可
内部ベクトルを抽出できるLLM	Llama，Mistral，Falcon，Gemma	可能（自前実装）	Hugging Faceなどからhidden_statesを取得
API非公開LLM（生成専用）	GPT-4（API），Claude，Gemini	不可（APIは非公開）	Embedding層が外部に公開されていない
RNN系，古いLLM	GPT-2，XLNet，ElMoなど	理論上可能だが実用的でない	出力が安定しづらい，文脈保持が弱い

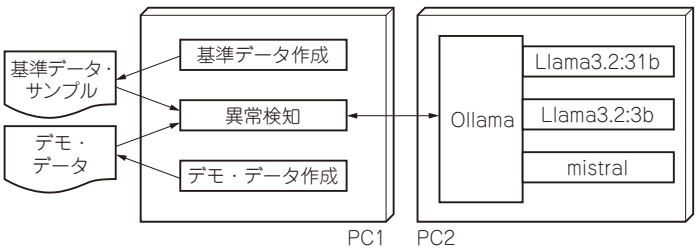

図1
実験のシステム構成

● 正常データとどれだけ意味的に似ているかで判断

Embedding（埋め込み）とは，テキストやログなどのデータを，その意味的特徴を数値的に表現した高次元ベクトルに変換する技術です．これを用いて，正常データと分析対象データをベクトル化し，それらのコサイン類似度を計算することで，両者がどれだけ意味的に似ているかを定量的に評価できます．

コサイン類似度は，Embeddingベクトル間の類似性を測る尺度であり，2つのベクトルがどれだけ同じ方向を向いているかを－1（完全に逆方向）から1（完全に同方向）の値で表します．距離ではなく角度に基づく指標です．

具体例として，単語AのEmbeddingベクトルを$\mathbf{v}_A$，単語Bのベクトルを$\mathbf{v}_B$とします．$\mathbf{v}_A$が「猫」，$\mathbf{v}_B$が「子猫」を表す場合，両者は意味的に近いためベクトルの方向も近く，コサイン類似度は高い値（例：0.9）になります．一方，$\mathbf{v}_B$が「車」を表す場合は意味が異なるため，コサイン類似度は低い値（例：0.2）になります．

このようにEmbeddingはデータの意味を数値ベクトルとして表現し，コサイン類似度はその意味的な近さを定量化する手法です．

従来のしきい値判定やキーワード検出とは異なり，Embeddingでは複数の要素が組み合わさった潜在的なパターンのずれを検出できる点が大きな特徴です．

● 主なLLMのEmbedding対応状況

LLMによるEmbeddingと類似度を使った手法は，単語単位だけでなく，文章全体や時系列データにも応用できます．そのため，センサ値やログ系列などにも柔軟に対応でき，ローカル環境での軽量な異常検知システムの構築に適しています．**表1**に主なLLMのEmbedding対応状況を示します．

実験のシステム構成

● 実験環境

実験環境を**図1**に示します．使用したPCは第3章と同じです．