Overcoming Vision Language Model Challenges in Diagram Understanding: A Proof-of-Concept with XML-Driven Large Language Models Solutions

Abstract

図は、ビジネス ドキュメント内の複雑な関係やプロセスを視覚的に伝える上で重要な役割を果たします。さまざまな画像理解タスク用の視覚言語モデル (VLM) の最近の進歩にもかかわらず、図に描かれた構造と関係を正確に識別して抽出することは、依然として大きな課題となっています。この研究では、VLM の視覚認識機能への依存を回避するテキスト主導のアプローチを提案することで、これらの課題に対処します。代わりに、図の要素 (図形、線、注釈など) がテキスト メタデータとして保存されている、xlsx、pptx、docx などの編集可能なソース ファイルを使用します。概念実証では、xlsx ベースのシステム設計ドキュメントから図の情報を抽出し、抽出した図形データを大規模言語モデル (LLM) のテキスト入力に変換しました。このアプローチにより、LLM は画像ベースの処理のボトルネックなしで関係を分析し、ビジネス指向の質問に対する応答を生成できました。 VLM ベースの方法との実験的比較により、提案されたテキスト駆動型フレームワークは、図の構造の詳細な理解を必要とする質問に対して、より正確な回答を生み出すことが実証されました。この研究で得られた結果は、テストされた .xlsx ファイルに限定されず、Office pptx や docx 形式などのソース ファイルを含む他のドキュメントの図にも拡張できます。これらの調査結果は、元のソース ファイルから直接テキストを抽出することで VLM 制約を回避できることを強調しています。LLM を通じて堅牢な図の理解を可能にすることで、私たちの方法は、実際のビジネス シナリオでワークフローの効率と情報分析を向上させる有望な道を提供します。関連コードは、この研究用に開発されたコア ライブラリを提供する https://github.com/galirage/spreadsheet-intelligence で入手できます。このライブラリを使用した実験コードは、https://github.com/galirage/XMLDriven-Diagram-Understanding にあります。

1 Introduction

情報を視覚的に整理して提示するダイアグラムは、文書内でテキストと視覚的表現を組み合わせる最も強力な方法の 1 つです。たとえば、システム アーキテクチャの設計図で要素間の関係を明示的に示したり、フローチャートを使用して複雑な手順やプロセスを段階的に説明したりするなど、ダイアグラムは情報伝達の効果的なツールとしてビジネス ドキュメントで広く活用されています。最近では、大規模言語モデル (LLM) (OpenAI 他、2024a、Anil 他、2023) の急速な発展により、テキスト情報を処理する方法がますます洗練されてきました。ドキュメントから得られた知識ベースと組み合わせた QA システム (Krishna ら、2024 年、Fleischer ら、2024 年)、自動化された要約とレポート生成による効率的なドキュメント作成と情報共有 (Thoppilan ら、2022 年、Liu ら、2025 年)、コンプライアンス検証タスクの自動化 (Cava と Tagarelli、2024 年、Kande ら、2024 年、Sollenberger ら、2024 年) などのアプリケーションにより、従来は手動で実行されていた複雑なタスクを自動化できるようになりました。これにより、ビジネス プロセスの効率化と標準化が大幅に加速されました。

さらに、画像を理解できる視覚言語モデル（VLM）の登場により（OpenAI et al., 2024b; noa, 2023; Fu et al., 2025; Google, 2024）、テキスト情報だけでなく図表も画像として処理することで、ワークフローの効率を高める可能性が高まっています。 VLM、特に独自のモデルは、写真やイラストなどの一般的な画像のキャプション生成や、視覚的な質問応答（VQA）などのさまざまなタスクで目覚ましい成果を上げています（OpenAI et al., 2024b; Google, 2024） さらに、図表などのより象徴的な視覚情報を分析する視覚推論タスクでも大きな進歩が遂げられており、テキストと図表を統合的に分析する自動化システムへの期待が高まっています（Lu et al., 2024; Zhang et al., 2024; Chen et al., 2024）。

こうした新たな可能性が広がる一方で、図に示された要素の構造や要素間の関係性を正確に把握・抽出することは容易ではなく、実際の業務に実装するにはまだ克服すべき課題が多く残っています。例えば、VLM は図形で構成された画像の幾何学的認識には限界があることが示されており (Kamoi et al., 2024; Zhang et al., 2024; Yue et al., 2024)、図を用いた質問応答 (QA) や推論タスクでは、幾何学的構造を正しくテキスト化する段階がボトルネックになることが報告されています (Ye et al., 2024)。質問応答 (QA) や図に関する推論などのタスクでは、幾何学的構造を正しくテキスト化する段階がボトルネックになることが報告されています (Ye et al., 2024)。さらに、重なり合う隣接する線や図形の装飾的な表現など、複雑な幾何学的プリミティブを含む画像は、VLM による誤認が発生しやすい (Rahmanzadehgervi et al., 2024) 傾向があり、既存の VLM は、実際の文書に見られるさまざまな配色や描画形式に対して十分に堅牢ではありません (Ye et al., 2024; Singh et al., 2024; Tannert et al., 2023)。さらに、VLM は、システムの設計図やフローチャートなどのエンティティ (グラフィック) の認識は比較的得意ですが、線で表された要素間の関係を正確に把握することは依然として困難です (Giledereli et al., 2024)。また、以前に学習した知識が入力視覚情報と一致しない場合、幻覚を起こす傾向があります (Mukhopadhyay et al., 2024; Giledereli et al., 2024)。これらの制限を考慮すると、ビジネス シナリオや視覚的に強化されたドキュメントを含むその他のコンテキストで言語モデルを効果的に活用するために、VLM の視覚機能の制約を緩和する方法を開発することが不可欠です。

本研究では、ダイアグラムを画像として処理するのではなく、ソースファイル内に格納されている形状情報を抽出して変換し、この情報を「LLM」を使用してテキストデータとして分析できるようにする新しいアプローチを提案します。ビジネスシナリオでは、ドキュメントはPDF形式（ISO、2016）に変換される前に、docx、xlsx、pptxなどの編集可能なソースファイル形式で存在することがよくあります。このソースファイルの実体はXMLであり、これを解析してダイアグラムを構成する形状、線、注釈などの他の要素に関する情報を直接抽出し、この情報をテキスト入力としてLLMに提供することで、視覚認識のボトルネックを回避し、ダイアグラムの包括的な理解と分析が可能になります。本研究では、日本のシステム開発ワークフローで一般的に使用されているExcelで作成された要件定義からシステム設計ドキュメントを準備しました。これらのドキュメントを使用して、いくつかのビジネス関連の質問を提示して評価を行い、提案手法とVLMベースのアプローチを比較しました。その結果、VLMは特定の質問に正確に答えることができなかったが、提案手法は正確な答えを提供することが確認されました。この研究は、XML などのソース ファイル処理による分析が、VLM におけるダイアグラム理解の課題を克服する可能性を示しています。ビジネス コンテキストで LLM を活用した情報活用システムにおいて、ワークフローを改善し、効率性を高める可能性を強調しています。

図 1: 私たちのアプローチは、VLM の視覚認識に依存せず、LLM をテキストとして入力して分析し、ソース ファイルに保存されている基礎となるグラフィック情報を参照してダイアグラム イメージをレンダリングします。

2 Related Works

2.1 Visual Recognition Limitation of VLM in Diagram Understanding

VLM の開発においては、さまざまな検証研究が報告されるにつれて、その視覚認識性能における課題が明らかになっています。たとえば、VLM は図形で構成された画像の幾何学的認識には限界があることが示されており (Kamoi et al. (2024); Zhang et al. (2024); Yue et al. (2024))、図表を使用した質問応答 (QA) や推論タスクでは、幾何学的構造を正しくテキスト化する段階がボトルネックになることが報告されています (Ye et al. (2024))。質問応答 (QA) や図表に関する推論などのタスクでは、幾何学的構造を正しくテキスト化する段階がボトルネックになることが報告されています (Ye et al. (2024))。また、重なり合う隣接する線や図形の装飾的な表現など、複雑な幾何学的プリミティブを含む画像は、VLM による誤認が発生しやすい (Rahmanzadehgervi et al.)。 (2024)、既存の VLM は、実際のドキュメントに見られるさまざまな配色や描画形式に対して十分に堅牢ではありません Ye et al. (2024); Singh et al. (2024); Tannert et al. (2023)。さらに、VLM はシステム設計図やフローチャートなどのエンティティ (グラフィック) の認識は比較的得意ですが、線で表された要素間の関係性を正確に把握することは依然として難しく (Giledereli et al. (2024))、以前に学習した知識が入力視覚情報と一致しない場合に幻覚を起こす傾向があります (Mukhopadhyay et al. (2024); Giledereli et al. (2024))。これらの制限を考慮すると、ビジネス シナリオや視覚的に強化されたドキュメントを含むその他のコンテキストで言語モデルを効果的に活用するためには、本研究で提示した方法のような、VLM の視覚機能の制約を緩和する方法の開発が不可欠です。

2.2 The Role of Source File Information in Improving Visual Recognition and Understanding by VLMs and LLMs

人間が通常閲覧するソースファイルのレンダリング形式を VLM や LLM に理解させる際に、ソースファイル情報を入力として利用して理解を容易にする例がいくつかあります。VLM ベースの GUI 理解の分野では、画面のスクリーンショットを画像として入力する一方で、ウィジェットやボタンのレイアウトや位置情報、テキストによる説明をテキスト入力として同時に提供することで、認識のパフォーマンスが向上します You et al. (2024); Li et al. (2024)。また、HTML や docx などのテキスト情報や文書構造などのメタデータを含むソースをモデルに直接入力する手法は、VLM における文書理解および分析タスクの精度を向上させることが報告されています Xu et al. (2020); Huang et al. (2022); Tan et al. (2024)。ダイアグラム理解の文脈では、ソースファイルからのメタデータ情報を利用することの利点を明示的に示す研究はありませんが、多くの結果がその可能性を示唆しています。たとえば、フローチャート理解のタスクを分析した研究では、フローチャート画像から視覚認識によってマーメイド表記を生成する段階と、フローチャートに関する質問に答える段階を分離し、画像からマーメイドへの変換にボトルネックがあることが実証されました (Ye et al. (2024); Pan et al. (2024))。また、正しいトポロジが理解されれば、フローチャートに関する質問に答えることは難しくないことも示されました。これらの調査結果は、前のセクションで強調した VLM の視覚認識の限界に関する複数のレポートと相まって、ドキュメント内のダイアグラム理解においてソース ファイル情報を活用することの有効性を裏付けており、この調査結果と一致しています。

3 Methodology

3.1 Diagram Information Parsing from Xlsx

前述のように、本研究では、文書を視覚情報として扱うことなく、ソースファイルに格納されたダイアグラム情報を抽出して変換することで、LLMがダイアグラムを理解・分析し、関連する質問に答えることができるワークフローを提案します。抽出された情報は適切にフォーマットされ、テキストに変換されます。

まず、文書を構成する情報はソースファイル内にXML形式で保存されているため、このXMLを解析して図に関する情報を抽出するライブラリを開発した。本研究では、Microsoft Excelで作成された.xlsxファイルとして保存されたシステム設計図を対象とした（図2）。Officeソフトウェアで作成された文書は世界的に普及しており、LLMでこれらの文書を解析できることは非常に望ましい。しかし、LangChain UnstructuredやAzure Document Intelligenceなどのデータローダーは利用可能であるが、これらのローダーは主にテキストデータの読み込みに限定されており、視覚的な要素をテキストに解析するものではない。

また、日本では要件定義書やシステム設計図の作成にExcelが広く利用されており、表データだけでなくExcelから図表を抽出したいというニーズも高い。システム設計図は図形と線、そして図形や線が表す概念を説明するテキストから構成されており、図形が実体、線が関係性を表す代表的な図表である。そのため、本研究では特に.xlsxファイルに保存されたシステム設計図を対象とした。

これを実現するために、.xlsx 形式のソース ファイルを解析し、ダイアグラムに関連する図形情報を抽出するライブラリを開発しました (https://github.com/galirage/spreadsheet-intelligence)。このライブラリを使用して抽出したデータを図 3 に示します。.xlsx を含む Office ドキュメントのソース ファイルは、基本的に zip 形式に圧縮された XML ファイルです。XML を解凍して読み取ると、ドキュメントを構成する要素を取得できます。XML には、テキスト、図形、テキスト ボックスなどのユーザーが配置したオブジェクトに関する情報と、描画位置、回転、テーマ、色、図形の書式設定、テキストの書式設定などの編集可能な属性が含まれています。

Excelで作成したシステム設計図の例を図2に示します。この図はテキストボックス、四角形、直線コネクタ、曲がったコネクタなどのオブジェクトで構成されています。図の構成要素として、これら4種類のオブジェクトに関連する情報を抽出しました。

図 2: この研究で使用した図。長方形、テキスト ボックス、直線コネクタ、曲がったコネクタを使用して Excel で描画されたシステムの設計図。

3.2 Parsing and Transforming Diagram Elements for Enhanced Analysis

次に、抽出した図形情報を LLM が理解しやすい形式に変換しました。XML から抽出した情報は、そのままでは十分な分析精度が得られません。そのため、長方形、テキスト ボックス、直線コネクタ、曲がったコネクタについては、図に関連する情報のみを抽出することにしました。図形とコネクタについては、まず座標情報を抽出しました。XML で提供される図形とコネクタの座標情報には、回転、反転、拡大縮小などの変換は反映されません。これに対処するために、これらすべての要素を組み込む変換プロセスを適用し、レンダリングされた図の外観を正確に表す値を確保しました。また、塗りつぶし、境界線、線の色情報については、ソフトウェア固有のカラー テーマ名を使用して色を指定した場合や、XML データがレンダリングされた状態に直接対応していない場合も、変換によって処理しました。図では、エンティティを表す図形をコネクタで接続することで関係を表現します。コネクタと図形の接続情報は、図の作成者が編集時に明示的に設定した場合は保存されますが、切断された図形とコネクタの関係も図で表現できます。このような場合、XML から直接図形とコネクタの接続を判断することはできません。そのため、ソース ファイルの接続情報に頼るのではなく、コネクタの座標データに基づいて、各コネクタの端点の方向を 4つの基本方向にデコードすることにしました。これにより、コネクタがどの方向を向いているかを把握し、各コネクタに接続されている図形を識別できます。また、Office ソフトウェアの図形はグループ化して図形のコレクションを表すことができますが、これらのグループの作成方法はダイアグラムの作成者によって異なるため、このプロセスではグループ情報は入力データとして利用されませんでした。次に、解析された情報は、図3 に示すように、図形 (テキスト ボックスと四角形の両方を含む) とコネクタ (直線コネクタと曲がったコネクタの両方を含む) ごとにグループ化された属性を持つ JSON 形式に整理されました。

図 3: 左側は xlsx ファイルから抽出したシェイプ データの XML 抜粋です。生の数値とエイリアス文字列のため解釈が困難です。また、複数のファイルにまたがる過剰な散在した情報も含まれています。右側は、コネクタとシェイプに関する意味のある情報を解析、変換、要約する JSON 形式です。ダイアグラムの理解と LLM 入力用に最適化されています。

4 Results

4.1 Entity Understanding Using the XML-driven Approach

XML から抽出したダイアグラム情報を用いてダイアグラムを解析し、それがシステム設計図の理解にどのように貢献するかを検証する例を示す。まず、図 2 に示すシステム設計図におけるコンポーネントやテキストボックスなどの各エンティティの理解に着目する。これには、長方形とテキストボックスで構成されるコンポーネント、テキストボックスを使用したコネクタの注釈情報、URL などのエントリポイントを表すテキストボックスが含まれる。前手法に基づいて JSON 形式に変換されたダイアグラムと、各属性を説明するプロンプトを使用して、コンポーネント、注釈、その他のテキストボックスを分類してリストするようにシステムに指示した。このとき、アイコン画像は入力から除外されている。JSON 形式には、長方形とテキストボックスの 4 辺の位置 (「左」、「右」、「下」、「上」) などの情報が含まれている。テキストボックスの場合は、ボックス内に含まれる文字列も記録される。また、塗りつぶしの色、境界線の色、形状が丸みのある長方形か普通の長方形かなどの属性が付与されている。図4は、LLMに図中のエンティティを理解・列挙するように指示した結果である。この結果、すべてのコンポーネントと注釈が正しく列挙され、注釈のテキストボックスも正確にリストアップされた。また、この入力データにはグループ化情報が含まれていないため、JSONに含まれる位置情報などを利用して、コンポーネント名を表すテキストボックスと周囲の四角形のペアを決定する必要があった。各コンポーネントを構成する図形のIDも出力したところ、システムが正しく図形を識別・グループ化していることが確認できた。

図4：図中のエンティティ理解における提案手法（テキスト駆動）とVLM（画像入力）の出力の比較。上図は入力システムの設計図です。各図形に割り当てられたIDは、テキストで図形情報を渡す際に提案手法が割り当てたIDを示しています。どちらの手法もエンティティ理解は漏れなく正しく行われており、XML駆動アプローチではグループ構造情報なしでテキストボックスと丸い四角形で構成されるコンポーネントを理解できることがわかります。

4.2 Relation Understanding Using the XML-driven Approach

次に、コンポーネント間の関係を表すコネクタの情報が正しく解析できるかを検証しました。この例では、直線コネクタや曲線ジグザグコネクタなど、形状の異なるコネクタが含まれています。このケースでは、先に解析した形状情報の出力（コンポーネントと注釈の詳細）とJSON形式の図の両方をプロンプトとして含めました。すべてのコネクタを出力するように指示し、それらが接続する2つのターゲットコンポーネントと、該当する場合は関連する注釈を指定しました。形状情報の解析結果には、各コンポーネントの名前と、コンポーネントが配置されている領域の位置情報が含まれていました。注釈テキストボックスの情報も同様です。コネクタのJSON形式の情報には、直線コネクタとジグザグコネクタの両方の始点と終点の（x、y）座標と、各終点が示す方向が含まれています。この情報を使用して、GPT-4にデータを解析するように指示しました。図5に結果を示します。その結果、システムはすべてのコネクタのターゲットを正しく識別し、抽出しました。曲がったコネクタ、まっすぐなコネクタ、テキストボックスとコンポーネントを接続するID:7など、コンポーネント間コネクタではないコネクタでもコネクタが正しく検出されました。また、「Key Vault」や「Microsoft Entra」など、コネクタが存在しない領域にも、何の疑問もなく回答されました。

図5: ダイアグラム内の関係性を理解するための提案手法 (XML 駆動) と VLM (画像入力) の出力比較。上図は各コネクタに割り当てられた ID を示しています。XML 駆動アプローチではコネクタの理解が漏れなく正しく行われているのに対し、画像入力では、黄色: コネクタの曲がりを検出していない可能性があります、赤: 存在しない関係を誤認し、オブジェクトの 1 つがテキスト ボックスである ID:7 のコネクタを検出しています。画像入力は黄色: コネクタの曲がりを検出していない可能性があります。

4.3 Comparison with VLM

GPT-4o の画像入力を使用して、同じシステム設計図を分析しました。まず、画像入力からすべてのコンポーネントと注釈をリストするタスクを実行しました。図 4 に結果を示します。その結果、すべてのコンポーネント名が漏れなく網羅的にリストされました。同様に、注釈テキストも欠落することなく完全にキャプチャされました。次に、提案手法で使用されるものと同様の指示を与えて、関係を分析する能力をテストしました。図 5 に結果を示します。直線コネクタは 2つのターゲットをリンクしていると正しく識別されましたが、曲線コネクタでは錯覚が見られました。たとえば、実際には Azure Cognitive Search と Azure App Service をリンクする曲線コネクタは、曲線の角の近くにあった Azure AI Document Intelligence に接続すると誤認識されました。また、Azure App Service と URL テキスト ボックス間のコネクタが出力にありませんでした。さらに、実際にはコネクタが存在しない「Key Vault」や「Microsoft Entra」などの無関係なコンポーネントと Azure App Service 間のコネクタが誤って認識されました。私たちが提案した XML 駆動型の分析と比較して、視覚認識に依存せずにソース ファイルから直接抽出された図情報を処理することで、視覚認識に固有の検出エラーの影響を受けずに LLM が図を理解できることが証明されました。

5 Conclusion

本論文では、視覚認識に頼るのではなく、Excel の XML などのソース ファイルから抽出した情報によって駆動される LLM を使用したダイアグラム理解方法を構築するというアイデアを提案します。VLM を使用する従来の画像駆動型アプローチは、本研究の幻覚例で実証されているように視覚機能が不十分であり、ビジネス コンテキストなどの実際のアプリケーションには依然として課題があることを示しています。これに対して、ダイアグラムの作成に使用された元のソース ファイルから抽出された形状データを活用する XML 駆動型ダイアグラム理解ソリューションを提示し、その実用的な可能性を強調しました。

しかし、本研究は特定のシステム設計図を用いた予備実験に限定されているため、統計的厳密性や評価データの一般化可能性を十分に検討していません。より広範囲で本手法の有効性を検証するには、大規模で多様なデータセットでのさらなるテストが必要です。ただし、本研究で使用した Excel ドキュメントの XML ソース ファイルを解析するために開発したライブラリは、オープンソース ソフトウェアとして公開されており、再現性と適用性が保証されています。これにより、研究コミュニティはさまざまなシナリオで本手法を再現し、改良して、さらに開発を進めることができます。

統計的手法が十分に厳密ではないことや、大規模データの一般化可能性が十分に保証されていないことなどによる制約はあるものの、ライブラリをオープンソース化することで、さらなる共同研究や産業応用が促進されると期待しています。この取り組みが、言語モデルを用いたダイアグラム理解技術の実用化を促し、ビジネス分野における要件定義やシステム設計分析の精度を高め、ひいては日本のシステム開発現場の業務効率化に貢献することを期待しています。