

生成 AI による Excel 文書解釈プロセスの二段階モデル

～抽出到達性と意味推論性に基づく構造的整理～

A Two-Stage Model for Interpreting Excel Documents with Generative AI

～ Structural Organization Based on Extractability and Semantic Inferability ～

リーダー：山本 文彦 (T I S 株式会社)

主 査：石川 冬樹 (国立情報学研究所)

副 主 査：徳本 晋 (富士通株式会社)

アドバイザー：栗田 太郎 (フリー株式会社)

研究概要

本研究は、企業実務において広く利用されている Excel 文書を生成 AI で分析・活用する際に生じる解釈結果の差異について、その要因を体系的に整理することを目的とする。近年、大規模言語モデル (LLM) を用いた業務文書理解の試みが進んでいるが、Excel 文書を対象とした場合、生成 AI の出力が文書作成者の意図と乖離する事例が少なくない。これらの乖離はしばしば生成 AI の性能として一括りに議論されるが、入力前処理に起因する情報欠落と、生成 AI 自身の意味推論挙動とが十分に区別されていない点が課題である。

そこで本研究では、Excel 文書が生成 AI によって解釈される過程を、「情報が生成 AI に到達するまでの抽出到達段階」と、「到達した情報に対して意味的な解釈・補完が行われる意味推論段階」という二つの段階に分けて捉える二段階モデルを仮説的枠組みとして導入した。この枠組みに基づき、抽出方式の違いが生成 AI の認識結果に与える影響と、意味推論段階における構造理解の特性を切り分けて検証した。

具体的には、Excel ファイル解析ライブラリ (openpyxl) による一般的な抽出方式と、Excel ファイルの内部構造 (OOXML) を直接解析する拡張抽出方式を用いて Excel 文書から情報を抽出し、JSON 形式で生成 AI に入力する実験環境を構築した。同一の生成 AI モデルに対して抽出方式のみを変更して入力することで、画像・図形・グラフなどの非セル要素が抽出到達段階で推論対象となるか否かを比較するとともに、抽出されたテキスト情報に対する意味推論挙動を観測した。

その結果、非セル要素は抽出段階で JSON に含めなければ生成 AI の推論対象とならない一方、JSON に含めたテキストについては、罫線や結合セルといった明示的な表構造情報が与えられていなくても、見出し語と値語の意味的一貫性が高い場合には、生成 AI が表構造を補完的に推論する挙動が確認された。また、配置や座標が同一であっても、これらのテキストの意味的一貫性を失わせると構造認識が成立しないことから、意味推論段階における理解は視覚的装飾よりもテキスト内容に強く依存していることが示唆された。

これらの知見から、本研究は、Excel 文書の生成 AI 可読性を「抽出到達性」と「意味推論性」という二つの観点に分けて整理する必要性を示すとともに、抽出方式の前提に応じた Excel 文書設計の簡易ガイドラインを提示した。本研究の枠組みは、生成 AI の性能評価に留まらず、実務における Excel 文書設計と生成 AI 活用を接続するための基礎的視点を提供するものである。

1. はじめに

1.1 背景

近年、生成 AI を業務文書の理解や分析に活用する取り組みが広がっており、企業内に蓄積された既存の業務文書資産を対象とした活用にも注目が集まっている^[1]。多くの企業において、要

求や設計に関する資料，進捗や変更を管理するための文書などが，日常的な業務遂行の中で作成・蓄積されている。

これらの業務文書の多くは，Excel を含む Office 系文書として管理されていることが多く^[2]，特に Excel は表形式による情報整理や更新の容易さから，業務の現場において広く利用されてきた。

一方，近年の大規模言語モデル（LLM）は，Transformer アーキテクチャに基づいて設計されており，入力はトークン化されたテキストとして与えられることを前提としている。そのため，表形式データを扱う場合であっても，モデル内部では二次元的な構造を直接処理するのではなく，何らかの方法で線形化されたテキスト列として入力されるのが一般的である。このため，Excel 文書を生成 AI で扱う際には，ファイルをそのまま解釈させるのではなく，セル値や属性情報を抽出し，JSON や CSV 等の構造化テキストとして入力する前処理が広く採られている^{[3][4][5]}。

しかし，この変換過程において，Excel 文書に含まれるすべての情報が必ずしも生成 AI に入力されるとは限らない。セル値のように比較的容易に抽出できる情報がある一方で，装飾，図形，グラフといった要素は，前処理の設計や実装によっては入力から失われる可能性がある。その結果，生成 AI による解釈や評価が，元の Excel 文書の意図や内容と乖離する場合がある。さらに，生成 AI の出力結果が期待と異なる場合，その原因が，入力段階で情報が欠落していることによるものなのか，それとも入力された情報に対する生成 AI の解釈・推論に起因するものなのか，十分に切り分けられないまま議論されることが多い。このような状況では，Excel 文書を生成 AI で活用する際の設計指針や改善方策を体系的に整理することが難しい。

以上の背景から，Excel 文書を生成 AI で扱う際には，入力前処理による情報の到達性と，到達した情報に対する生成 AI の解釈・推論の挙動とを区別して整理することが必要である。

1.2 本研究の目的

本研究の目的は，Excel 文書を生成 AI で扱う際の解釈結果の差異について，その要因を体系的に整理することである。

具体的には，Excel 文書が入力されてから解釈結果が得られるまでの過程を，「Excel 文書からの情報を抽出し生成 AI に到達させる挙動」と「生成 AI に到達した情報に対する意味推論の挙動」という二つの観点に分けて捉え，それぞれが解釈結果に与える影響を明らかにする。

2. AI 可読性の二段階モデル

本章では，第 1 章で述べた問題意識を踏まえ，生成 AI による Excel 文書理解の過程を，「抽出到達段階」と「意味推論段階」の二つの段階に分けて捉える枠組みとして，AI 可読性の二段階モデルを導入する。以降の検証では，本章で示す二段階モデルを前提として，実験設計および評価を行う。

2.1 抽出到達段階（Step1）

第一段階は，Excel 文書に記載された情報が，抽出処理を経て JSON として生成 AI に渡される段階である。

この段階において JSON に含めることができなかった情報は，生成 AI にとっては存在しないものとして扱われ，以降の推論対象とはならない。

2.2 意味推論段階（Step2）

第二段階は，JSON に含まれたテキスト情報をもとに，生成 AI が意味的な関係性や構造を推論・補完する段階である。

3. 実験

3.1 実験の全体構成

本検証では、第2章で示した「AI 可読性の二段階モデル」を前提とし、Excel 文書が生成 AI に入力されてから解釈結果が得られるまでの過程を、抽出到達段階 (Step1) と意味推論段階 (Step2) に切り分けて観測するための実験環境を構築した。

具体的には、異なる抽出方式によって得られた入力データを同一の生成 AI モデルに与え、抽出方式の違いが、Excel 文書中の主要要素の認識および解釈結果にどのような影響を与えるかを比較・評価した。これにより、情報が生成 AI に到達しないことによる影響と、到達した情報に対する生成 AI の推論挙動を切り分けて観測することを試みた。

以下では、

(1) 情報抽出を担う抽出レイヤと検証対象の Excel ファイル

(2) 生成 AI への入力データの表現形式

(3) 意味推論を担う生成 AI 基盤

の順に、実験環境の詳細を説明する。

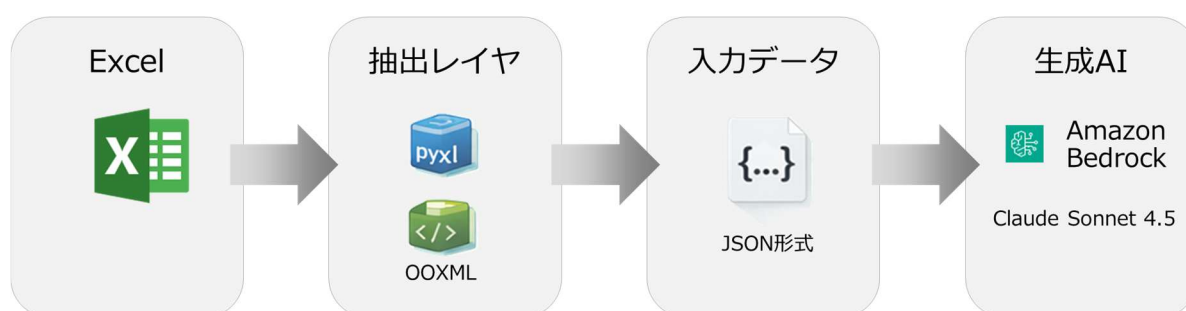


図1 AI 可読性の二段階モデルに基づく実験環境の構成

(1) 抽出レイヤと検証対象

本検証では、抽出レイヤに入力する検証対象として、計画書、進捗管理表や課題管理表など、実務で一般的に用いられる Excel 管理資料をもとに、文書構成を簡略化・抽象化しつつ、実務上の記述の癖や構成上の特徴を意図的に含めた検証用 Excel ファイルを用いた。

Excel ファイルから生成 AI への入力用データを生成する方法としては、大きく分けて、Excel ファイルを操作するアプリケーションやライブラリを介して情報を取得する方法と、Excel ファイルの内部構造を直接解析して情報を取得する方法の二つが考えられる。本検証では、これら二つの考え方に基づく抽出方式として、以下の二つの抽出レイヤを用いた。

・アプリケーション／ライブラリを介した情報抽出

本研究では、Excel ファイル解析ライブラリである openpyxl を用いた。openpyxl 等のライブラリを用いた抽出では、セル値や基本的な表構造を比較的容易に取得できる一方で、画像、図形、グラフ等のセル以外の要素（以下、非セル要素）については、取得に制約がある。

・Excel ファイルの内部構造を直接解析する情報抽出

本研究では、Excel ファイルの内部構造である OOXML を直接解析する自作処理を用いた。この方式では、非セル要素についても構造的に取得することが可能であるが、実装コストが高く、また解析対象とする要素の網羅性に注意が必要となる。

(2) 入力データ表現

本研究では、表構造の入力表現として、行単位や列単位でデータを配列化する形式は用いず、各セルの位置情報（行番号・列記号）とセル内容を明示的に対応付けた記述方式を採用した。具体的には、「[セル(行,列)] 内容」の形式でセル位置とテキストを列挙し、シート単位で JSON に格納している。この表現では、罫線やセル結合といった明示的な表構造情報を含めていない

が、生成 AI に入力した際、セル位置とテキスト内容の対応関係をもとに、表として解釈・推論される挙動が確認されている。

本研究では、このようなセル位置に基づく線形化表現を前提として、抽出方式の違いおよび意味推論挙動の差異を比較した。また、画像・図形・グラフといった非セル要素については、OOXML を直接解析することで、要素種別および配置情報を取得し、JSON にメタデータとして付与した。具体的な入力表現例については付録 A に示す

(3) 生成 AI 基盤（意味推論段階）

意味推論段階（Step2）における生成 AI として、本検証では Amazon Bedrock 上で提供される Claude Sonnet 4.5 を用いた。

Amazon Bedrock を利用することで、モデル選択や実行環境をクラウド基盤上で統一でき、企業システムへの適用を前提とした再現性の高い検証が可能となる。また、Claude Sonnet 4.5 は検証時における最先端モデルであり、高い推論性能と実用的なビジネス文書生成能力を有するので、本研究で対象とする Excel 管理資料の構造推論評価に適していると判断した。

3.2 検証結果（抽出レイヤ別）

本検証では、3.1 節で示した二つの抽出レイヤを用い、同一の Excel 文書を対象として抽出処理を行った。抽出結果として得られた JSON を目視で確認し、Excel 文書に含まれる各要素が生成 AI の推論対象となり得る形で取得されているかを評価した。

本節では、この評価結果に基づき、抽出レイヤごとに各要素の取得可否を整理する。主要な Excel 要素についての取得可否は、付録 B「付表 1 抽出レイヤ別の取得可否」に示す。

【抽出レイヤ別の検証結果】

一般的なライブラリ（openpyxl）を用いた場合、セル値や単純な表構造については概ね取得可能であり、生成 AI に入力された後、意味推論段階において表構造を補完する挙動が観測された。一方、セル結合、罫線、背景色といった装飾情報については、本研究時点の実装では JSON に含めることができず、生成 AI による推論対象とはならなかった。

また、画像・図形・グラフ等のセル以外の要素（非セル要素）については、openpyxl による抽出では到達させることができなかったが、OOXML を直接解析する方式では、これらの要素を JSON に含めることが可能であった。これにより、同一の Excel 文書であっても、抽出方式の違いによって、生成 AI に到達する情報の範囲が大きく異なることが確認された。

以上の結果から、生成 AI の解釈結果に影響を与える前提条件として、どの情報が抽出到達段階で生成 AI に渡されているかが極めて重要であり、抽出方式の選択が意味推論段階の成否を左右する要因となることが確認できた。

3.3 表構造推論の観測（意味的一貫性の影響）

本節では、前節までに示した抽出到達段階（Step1）において生成 AI に入力された情報を対象として、意味推論段階（Step2）における生成 AI の解釈挙動を整理する。

具体的には、生成 AI に対して表の個数、見出し語と値語の対応関係、要素間の関係性を説明させ、その応答をもとに構造認識の成否を判定した。判定基準の詳細は付録 C に示す。

これにより、到達した情報をもとに、生成 AI がどのように構造や関係性を推論したかを観測した。

表構造推論における観測

本研究で用いた JSON 表現には、罫線やセル結合といった明示的な表構造情報を含めていない。しかし、複数のセルにわたり意味的な関連を持つテキストが配置されている場合、生成 AI

がセル間の関係性から見出し語と値語の対応を推定し、表構造として解釈する挙動が確認された。

例えば、列見出しや行見出しに相当する語彙と、その下位に配置された数値や属性値が意味的に整合している場合、生成 AI はこれらを同一の表構造に属する要素として認識し、見出し語と値語の対応関係を説明する応答を生成した。この挙動は、座標情報や配置情報が完全でない場合においても観測された。

意味的一貫性が推論に与える影響

本検証では、セル配置、行列構造、および JSON の記述形式を同一に保ったまま、セル内テキストのみを意図的に変更する条件を設定している。具体的には、意味を持つ語彙を含むテキストと、意味を持たない固定文字列とを置き換え、意味的一貫性のみが生成 AI の構造認識に与える影響を比較した。

この条件下では、同一の座標配置および JSON 構造を維持しているにもかかわらず、セル内テキストを固定文字列に置き換えた場合、生成 AI は表構造を正しく認識できず、要素間の関係性の推定に失敗することが確認された。なお、本研究では、見出し語と値語の対応説明が成立しない場合、または表の個数や関係性を誤認した場合を「失敗」と定義した。

一方で、意図的にセル配置や座標関係を崩した場合であっても、見出し語と値語の意味関係が明確であれば、生成 AI が表構造を補完的に推定する挙動も確認された。

これらの結果から、意味推論段階における構造認識は、視覚的配置や位置情報そのものよりも、テキスト内容の意味的一貫性に強く依存していることが示唆される。

この挙動を模式的に示したものが図 2 である。図 2 では、セル配置および構造が同一であるにもかかわらず、セル内テキストに意味のある語彙を用いた場合には生成 AI が表構造を認識できた一方、意味のない文字列を配置した場合には構造認識に失敗する様子を示している。

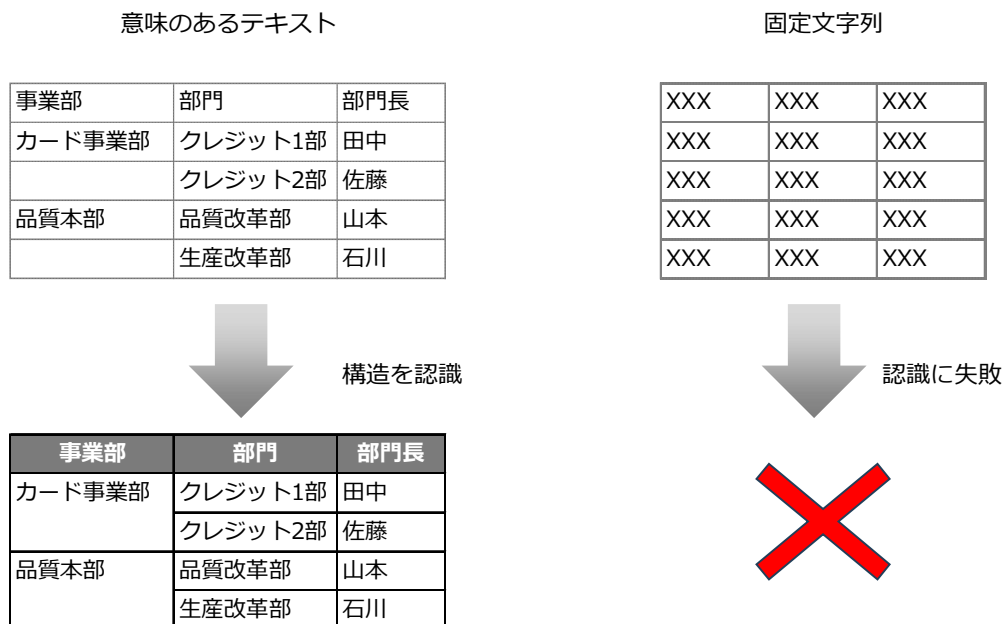


図 2 テキスト意味の有無による表構造認識の差異

意味推論段階の特性整理

以上の結果から、意味推論段階 (Step2) における生成 AI の挙動を、以下のように整理した。

- 抽出到達段階で生成 AI に入力された情報のみが推論対象となる
- 罫線等の明示的な表構造情報がなくても、テキストの意味的一貫性が高い場合、構造推論が成立する

- 意味的一貫性が失われた場合、同一の配置・構造を持っていても推論に失敗する

これらの結果は、生成 AI による Excel 文書の理解が、抽出段階で到達した情報の有無だけでなく、到達した情報の記述内容そのものに大きく依存していることを示している。

4. 実験結果の整理と考察

本章では、第3章で示した実験結果をもとに、Excel 文書を生成 AI で扱う際の情報認識過程について整理し、二段階モデルの観点から考察を行う。具体的には、抽出到達段階 (Step1) および意味推論段階 (Step2) において観測された挙動を総合し、本研究で明らかになった点と、今後の検討が必要な点を整理する。

4.1 抽出到達段階に関する整理と考察

抽出到達段階 (Step1) に関する実験結果から、生成 AI の解釈結果は、入力前処理においてどの情報が生成 AI に到達しているかに強く依存することが確認された。

openpyxl 等の一般的なライブラリを用いた抽出では、セル値や単純な表構造については概ね取得可能であり、生成 AI に入力された後、意味推論段階において表構造を補完する挙動が観測された。一方で、セル結合、罫線、背景色といった装飾情報や、画像・図形・グラフといった非セル要素については、正しく抽出できなかった。

これに対し、Excel ファイルの内部構造 (OOXML) を直接解析する方式では、非セル要素についても正しく取得でき、生成 AI に認識させることができた。

この結果、同一の Excel 文書であっても、抽出方式の違いによって生成 AI が参照可能な情報の範囲が大きく異なり、その後の解釈結果にも差が生じることが明確になった。

これらの結果から、生成 AI の出力が期待と異なる場合、その要因を生成 AI 自身の性能として一括りに評価するのではなく、抽出到達段階において対象情報が推論対象となっているか否かをまず確認する必要があることが示唆される。

4.2 意味推論段階に関する整理と考察

意味推論段階 (Step2) に関する実験では、抽出到達段階で生成 AI に入力されたテキスト情報をもとに、生成 AI がどのように構造や関係性を推論するかを観測した。

本研究で用いた JSON 表現には、罫線やセル結合といった明示的な表構造情報を含めていないにもかかわらず、セル内テキストの意味的一貫性が高い場合には、生成 AI が見出し語と値語の関係を推定し、表構造として解釈する挙動が確認された。

一方で、同一の配置・構造を維持したまま、セル内テキストを意味的な関連性を持たない文字列に置き換えた場合、生成 AI は表構造を正しく認識できず、要素間の関係性の推定に失敗した。このことから、意味推論段階における構造認識は、視覚的な配置や装飾よりも、テキスト内容の意味的一貫性に強く依存していることが示唆される。

また、セル配置や座標関係が一部崩れている場合であっても、見出し語と値語の意味関係が明確であれば、生成 AI が構造を補完的に推定する挙動も確認された。これらの結果は、生成 AI が入力情報を単なる配置情報としてではなく、意味的文脈を持つテキストとして解釈している可能性を示している。

4.3 二段階モデルの有効性に関する考察

以上の実験結果を総合すると、Excel 文書を生成 AI で扱う際の情報認識過程を、「抽出到達段階」と「意味推論段階」に分けて整理する二段階モデルは、解釈結果の差異を説明する枠組みとして有効であると考えられる。

抽出到達段階では、前処理の設計や抽出方式の違いによって、生成 AI が参照可能な情報の範囲が決定される。一方、意味推論段階では、到達した情報をもとに、生成 AI が意味的關係性や

構造を補完的に推論する。この二つの段階を区別して整理することで、「情報がそもそも入力されていないことによる解釈不能」と、「入力されているが意味推論に委ねられることによる解釈の揺らぎ」とを切り分けて議論することが可能となる。

なお、抽出処理と意味解釈を分離して扱う枠組み自体は一般的である。しかし、Excel 文書では抽出方式の違いにより生成 AI に到達する情報範囲が大きく変化する点に特徴がある。本研究は、この特性を「生成 AI への到達性」という観点で整理した点に特徴がある。

4.4 本研究の知見と残された課題

本研究を通じて、以下の点が明らかになった。

- 生成 AI の解釈結果は、抽出到達段階において生成 AI に入力された情報のみを前提として成立する。
- 明示的な表構造情報が与えられていなくても、セル内テキストの意味的一貫性が高い場合、生成 AI は表構造を補完的に推論する。
- 意味的一貫性が失われた場合、同一の配置・構造を持つ表であっても、生成 AI は構造認識に失敗する可能性が高い。

一方で、本研究では検証対象を代表的な構成に限定しており、実務で用いられるすべての Excel 表現や運用固有のフォーマットを網羅できていない。また、セル結合や罫線、背景色といった一部の装飾情報については、本研究時点では抽出・反映を行っておらず、評価対象から除外している。さらに、本検証は単一の生成 AI モデルを用いた定性的観測に基づくものであり、成功率やプロンプト依存性等の定量的評価は行っていない。

今後は、実務で利用されている管理資料をより多様に模したデータを用いた検証を行い、抽出到達性と意味推論性の寄与をより定量的に評価するとともに、他の生成 AI モデルとの比較検証を通じて本枠組みの一般性を検討することが課題である。また、本研究は Excel 文書を対象としているが、本二段階モデルが他の業務文書形式においても成立するかについては実証しておらず、今後の検討課題とする。加えて、OOXML 解析による拡張抽出についても、対象要素を段階的に拡充し、実装および設計指針の汎用化を進める必要がある。

5 実務への示唆：Excel 文書設計の簡易ガイドライン

本研究で整理した二段階モデルは、Excel 文書を生成 AI で活用する際の文書設計にも示唆を与える。本節では、実務での利用を想定し、抽出方式ごとに簡易的なガイドラインを示す。

(1) 一般的な API (openpyxl 等) を前提とする場合 (利用者向け)

- **セル結合への依存を避ける。**
結合セルでは抽出対象のセル値が欠落する可能性があり、構造が崩れる恐れがある。
- **罫線・背景色などの装飾で意味を表現しない。**
これらは抽出されない (または抽出対象外となりやすい) ため、生成 AI の推論対象に到達しない。
- **他シート参照に依存しない。**
参照先の値が抽出されないため、意図した情報が推論対象に含まれない。
- **画像・図形・グラフなどの非セル要素は原則として使用しない。**
これらは抽出されないものが多いため、重要情報の表現はセル内テキストで表現する。
- **見出し語を明確にし、語彙を統一する。**
同一概念の表記ゆれを抑え、意味的一貫性を高めることで、意味推論段階 (Step2) の安定性が向上する。

(2) OOXML 解析による拡張抽出を前提とする場合 (開発者・利用者の役割分担)

OOXML の内部表現は、見え方だけでなく、Excel における表現形式、操作手順、保存形式、およびバージョン差異等の要因により、複数のパターンが存在する。これらの要因に起因する OOXML 表現の差異は、Excel 文書を作成・利用する側が意識的に制御できるものではなく、利用者が特定の OOXML 表現形式を選択することは一般に困難である。

そのため、画像、図形、グラフといった非セル要素を生成 AI の推論対象とする場合には、開発者側において、抽出到達段階 (Step1) で対応する OOXML 表現の範囲を明示的に定義する必要がある。具体的には、画像であれば画像データおよび配置情報、図形であれば種別や囲み領域 (座標)、グラフであれば系列名、カテゴリ、数値 (キャッシュ) 等、意味推論に必要と考えられる情報を JSON として付与する設計を行う。

一方、利用者側は、開発者が想定する抽出結果が得られるよう、表現方法、操作手順、保存形式、および Excel バージョン等について指定された条件の範囲内で Excel 文書を作成する必要がある。これらの条件を満たさない場合、当該情報は生成 AI に到達せず、意味推論の対象とならない可能性がある点に留意する必要がある。

また、OOXML 解析による拡張抽出は、対応範囲の拡大に伴って実装コストが増大する。このため、開発者側は、実務上の重要度や利用頻度を考慮しつつ、解析対象とする OOXML 表現を段階的に拡充していく設計が望ましい。

なお、重要な判断や結論に関わる情報については、非セル要素に依存せず、セル内テキストとしても記載しておくことが、抽出到達性および意味推論の安定性の観点から無難である。

生成 AI の理解能力は今後向上する可能性があるが、解釈可能な情報が入力形式および前処理に依存するという構造自体は維持されることが考えられる。したがって、本研究で示した二段階モデルは、将来的にも基本的枠組みとして有効である。

6. おわりに

本研究では、Excel 文書を生成 AI で扱う際に生じる解釈結果の差異について、その要因を体系的に整理することを目的とし、情報認識過程を「抽出到達段階」と「意味推論段階」の二段階に分けて捉える枠組みを提示した。

実験を通じて、生成 AI の解釈結果は、抽出処理によって生成 AI に到達した情報のみを前提として成立すること、ならびに、到達した情報に対する意味推論においては、表構造や配置といった視覚的装飾よりも、セル内テキストの意味の一貫性が大きく影響することを確認した。これらの結果は、生成 AI の出力を評価・改善する際には、入力前処理と意味推論の挙動を切り分けて検討する必要があることを示している。

本研究で示した二段階モデルは、生成 AI の性能評価に留まらず、Excel 文書の設計および生成 AI 活用を接続するための基礎的な視点を提供するものである。今後は、より多様な実務資料を対象とした検証を進めるとともに、抽出方式および入力表現の拡張を通じて、本枠組みの実務適用可能性をさらに高めていくことが期待される。

参考文献

- [1] 独立行政法人 情報処理推進機構 (IPA), 「ソフトウェア動向調査 2024」, 2024.
- [2] 一般社団法人 日本情報システム・ユーザー協会 (JUAS), 「企業 IT 動向調査報告書 (IT2025 関連調査)」, 2025.
- [3] Li, J., et al., “TWT: Table with Written Text for Controlled Data-to-Text Generation,” Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2021.
- [4] OpenAI, “File Uploads FAQ,” OpenAI Help Center, 2023.
- [5] OpenAI, “Optimizing File Uploads in ChatGPT Enterprise,” OpenAI Help Center, 2023.

付録 A

本付録では、第 3.1 (2) 節で言及した JSON 入力表現の具体例を示す。

A.1 表データの JSON 表現形式 (基本例)

本研究では、Excel 管理資料に含まれる表データを、シート名およびセル座標と値の組として JSON 形式に変換し、生成 AI への入力データとして用いた。

```
{
  "type": "text",
  "text": "=== 【シート名】 Sheet1 ===\n[セル(1,A)] 部門名\n[セル(1,B)] 人数\n[セル(2,A)] 品質管理部\n[セル(2,B)] 15\n[セル(3,A)] 営業部\n[セル(3,B)] 20\n"
```

上記記述は Sheet1 に以下のように格納されたデータとして認識される。

	A 列	B 列
1	部門名	人数
2	品質管理部	15
3	営業部	20

A.2 画像：Base64+配置 (アンカー座標)

OOXML を直接解析することで、Excel ファイルを構成する ZIP 内の画像ファイルおよび、スプレッドシート上での配置情報 (アンカー座標) を取得できる。

これらを組み合わせた生成 AI への入力例は以下の通りである。

```
{
  "type": "text",
  "text": "<中略> [画像: (23,B)] image1.png\n "
}
{
  "type": "image",
  "source": {
    "type": "base64",
    "media_type": "image/png",
    "data": "iVBORw0KGgoAAAANSUHEUgAA...(省略)"
  }
}
```

上記記述はセル B23 の位置に image1.png があるものとして認識される。

このように、非セル要素についても配置情報を併せて与えることで、生成 AI の意味推論段階において、セル値との関係性を考慮した解釈が可能となる。

A.3 図形：種別+座標 (左上・右下セル)

図形については、OOXML 内の drawing.xml 等を解析することで、図形の種別およびスプレッドシート上での座標情報を取得できる。

これらを組み合わせた生成 AI への入力例は以下の通りである。

```
{
  "type": "text",
  "text": "<中略>--- 【DrawingML 図形検出結果】 ---\n[図形 roundRect] 範囲=セル(8,A) → セル(12,AC) 色=FF0000\n\n=== 【シート名】 メイン ===\n[セル(1,A)] 方眼図形混合
```

```

テスト\n[セル(2,A)] 組織図\n[セル(3,B)] 事業本部\n[セル(3,G)] 事業部\n[セル(3,N)] 部
門名\n[セル(3,U)] 部門町名\n[セル(3,Y)] 部員人数\n[セル(4,B)] クラウド事業本部\n[セル
(4,G)] インフラ事業部\n[セル(4,N)] データセンター運用部\n[セル(4,U)] 佐藤 健一\n[セル
(4,Y)] 45人\n...\n"
}

```

上記記述はセル A8~AC12 を囲むような矩形があるものとして認識される。

A.4 グラフ (ChartObject) : 系列名・カテゴリ・値 (キャッシュ)

グラフについては、OOXML 内の chart1.xml 等を解析することで、グラフに表示されている系列名、カテゴリ名、および数値データ、スプレッドシート上での配置情報 (アンカー座標) を取得できる。

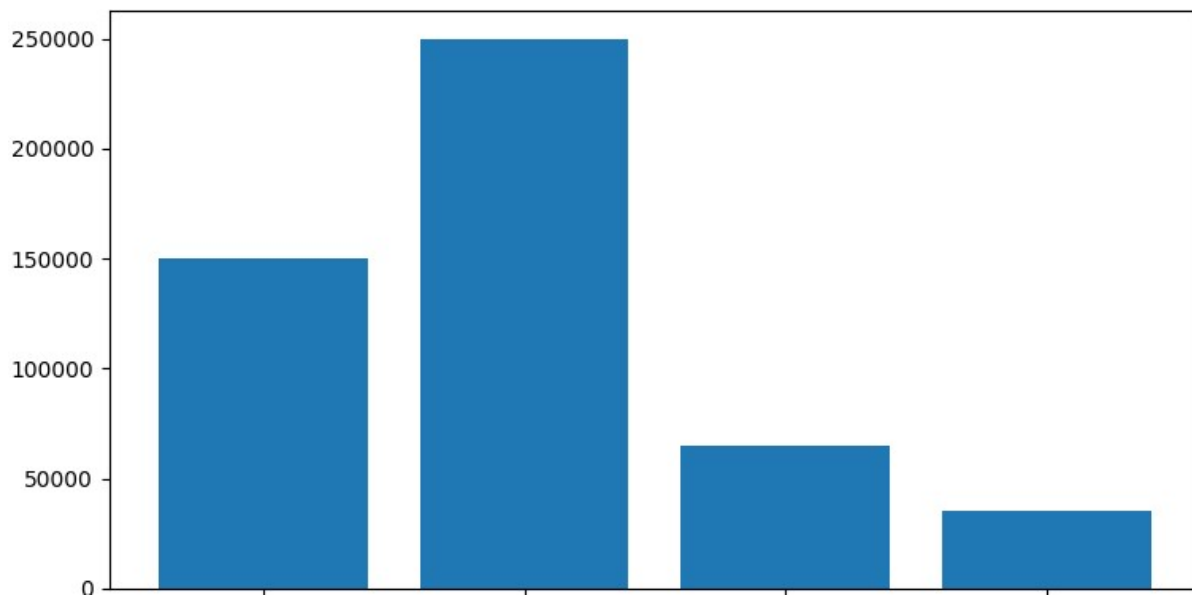
これらを組み合わせた生成 AI への入力例は以下の通りである。

```

{
  "type": "text",
  "text": "<中略> --- 【ChartObject (グラフ) キャッシュ値】 ---
\n[x1/charts/chart1.xml Series0] 名前='売上数'\n カテゴリ=['東京本部', '大阪本部',
'名古屋本部', '福岡本部']\n 値=[150000.0, 250000.0, 65000.0, 35000.0]\n\n=== 【シ
ート名】 <中略>"
}

```

本例では、下記のようなグラフあるものとして認識される。



付録 B

付表 1 抽出レイヤ別の取得可否

本表において OOXML 欄を「未検証」とした項目については、以下の理由による。

openpyxl により取得可能な要素については、抽出方式間で生成 AI への到達情報に差異が生じないため、比較検証の対象から除外した。

一方、openpyxl で取得できない要素については OOXML による抽出実装を検討したが、本研究は抽出到達段階における到達情報範囲の差異を明確化することを主目的としている。そのため、生成 AI の解釈結果に与える影響が大きいと考えられる画像・図形・グラフ等の非セル要素を優先して検証を行い、一部要素については実装および検証を範囲外とした。

検証ケース	認識結果		備考
	openpyxl	OOXML	
セルテキスト (セル結合なし)	○	未検証	openpyxl によりセル値および配置情報を取得できた。 到達した情報をもとに、生成 AI は表構造を補完的に推論した。
セルテキスト (セル結合あり)	×	未検証	openpyxl では結合セルに含まれる値の一部が取得できず、到達情報が欠落した。 OOXML による抽出は本研究時点では未実装である。
複数表・段組み	○	未検証	openpyxl では罫線情報は取得できなかったが、抽出されたテキストの並びをもとに、生成 AI は複数の表や段組み構成を推定した。
太字、イタリック、文字色等	△	未検証	openpyxl ではセル全体にかけられた書式は抽出できるが、セル内の文字単位でかけられた書式は抽出できなかった。 OOXML による抽出は本研究時点では未実装である。
罫線	×	未検証	openpyxl では罫線情報を取得できなかった。 OOXML による抽出は本研究時点では未実装である。
背景色	×	未検証	openpyxl では背景色情報を取得できなかった。 OOXML による抽出は本研究時点では未実装である。
コメント・メモ	○	未検証	openpyxl によりコメントおよび付与セルの位置情報を取得できた。 ただし、コメント／メモの区別はできない。
数式セル	○	未検証	数式セルについては、Excel 内部にキャッシュされた計算結果を openpyxl により取得できた。 取得した値は通常のセル値と同様に生成 AI へ到達させた。
他シート参照	×	未検証	openpyxl では他シートを参照する数式の参照先値を取得できなかった。 OOXML による抽出は本研究時点では未実装である。
画像	×	○	openpyxl では画像を取得できなかった。 一方、OOXML を直接解析する自作処理により、ZIP 内の画像ファイルおよび配置情報を取得し、JSON に含めて生成 AI へ入力した。 その結果、画像は生成 AI 側で OCR 処理により文字情報として解釈され、スプレッドシート上の値と関連付けて評価された。
図形	×	○	openpyxl では図形を取得できなかった。 OOXML を直接解析する自作処理により、図形の種別および座標情報を取得し、JSON に含めて生成 AI へ入力した。 その結果、図形で囲まれた範囲とセル値との関連付けが可能であった。
グラフ (ChartObject)	×	○	openpyxl ではグラフを取得できなかった。 Excel では、外部ブックを参照するグラフであっても、

			<p>グラフオブジェクト内に表示用の値が保持されている。 OOXML を直接解析する自作処理により、chart1.xml 等からこれらの値を取得し、 JSON に含めて生成 AI へ入力した結果、グラフ値として認識された。</p>
--	--	--	-----------------------------------------------------------------------------------------------------------------------------

付録 C 意味推論段階における評価基準および応答例

本研究では、生成 AI の応答について以下の基準により構造認識の成否を判定した。

- (1) 表の個数が入力構造と一致している
- (2) 見出し語と値語の対応関係を説明または列挙できている
- (3) 指定された範囲や対象を過不足なく抽出できている

上記条件を満たす場合を「成功」とし、いずれかを満たさない場合を「失敗」とした。
以下、成功例を3つ示す。

C1. 罫線で構成された複数表

【入力形式】

Excel 方眼上に罫線で4つの表を設計

【プロンプト】

「階層表現.xlsx に記載された表名を列挙して。さらに、すべての表に含まれる行の内容をすべて列挙してほしい。」

【応答（抜粋）】

- ・組織図
- ・部門別 KPI 評価
- ・注目プロジェクト
- ・部門別コスト配分

(各表の行内容をすべて列挙。)

【判定理由】

表の個数（4表）が正しく認識され、かつ各表の行内容が過不足なく列挙されたため成功と判定。

C2. 図形による範囲指定

【入力形式】

罫線で構成された表の一部を矩形図形でマーキング

【プロンプト】

「表に書かれている情報のうち、赤枠で囲まれている範囲の文字を抜き出してほしい。」

【応答（抜粋）】

AI 事業本部
AI 応用事業部
画像認識開発部

山田 悠人
30 人
(以下略)

【判定理由】

矩形で囲まれた範囲のテキストのみが抽出され、他領域の文字列は含まれていないため成功と判定。

C3. JPEG 画像貼付

【入力形式】

- ・表タイトル「注目プロジェクト」はシート上のセル文字列として存在
- ・表本体は画像として配置

【プロンプト】

「階層表現.xlsx に記載された表名を列挙せよ。」

【応答（抜粋）】

- ・注目プロジェクト
(注目プロジェクトに対応する画像内の表内容を列挙)

【判定理由】

セル上のタイトル文字列と画像内の表内容が対応付けられ、同一の表として認識されたため成功と判定。