令和7年度卒業論文

title1
title2

主任指導教員
栗原 聡

慶應義塾大学
理工学部管理工学科

田村早絵
62212594

提出日 2026年月日

概要

概 要

ここに概要を記述

第1章　序論

第2章　関連研究

第 2 章では，関連研究について述べる．具体的には，第 2.1 節において大規模言語モデル（Large Language Model，LLM）に関する研究を概説し，第 2.2 節ではシーングラフ生成に関連する研究を取り上げる．

2.1　大規模言語モデル

本節では, 大規模言語モデルの歴史と代表的なモデルについて述べる．

近年, 自然言語処理（Natural Language Processing; NLP）分野では, 大規模言語モデル（Large Language Model; LLM）と総称される技術群が飛躍的な進歩を遂げている．LLM は, 数千億から数兆に及ぶパラメータを有する巨大なディープラーニングモデルであり, 膨大なテキストデータを事前学習することで, 言語の文法構造, 語彙的意味的, 文脈依存性といった多様な知識を内部に獲得している．このようにして得られたモデルは, 自然言語を理解・生成する能力を備え, 単なる統計的言語モデルを超えた汎用的な言語知能として機能する．

LLM は多層のニューラルネットワークを基盤とし, 特に Transformer アーキテクチャ[1] に基づいて構築されている．この構造において中核的な役割を果たす自己注意機構（Self-Attention）は, 文中の単語間の関係を動的に捉えることで, 長距離依存関係を含む文脈の理解を可能にしている．代表的な LLM としては, OpenAI による GPT シリーズ [2][3][4][5][6] や Google の BERT[7] が挙げられ, 文章生成, 質問応答, 翻訳, コード生成など多様なタスクにおいて高い性能を示している．

さらに近年では, ChatGPT の登場により, LLM の社会的な影響力は飛躍的に拡大し

ている．学術研究や産業応用のみならず, 教育, ビジネス, 創作支援といった日常的な領域にも浸透しつつあり, 人と AI の新しい協働の形を実現する基盤技術として注目を集めている．

2.1.1 LLM の歴史

Zhao らのサーベイ論文 [8] によれば，言語モデルの発展は大きく 4 つの段階に整理できる．1990 年代から 2010 年代初頭にかけては，n-gram やマルコフモデルといった統計的言語モデルが主流であり，単語列の共起確率に基づいてテキストを生成・解析する枠組みが用いられた．これらのモデルは自然言語処理の基礎を築いたものの，語彙のスパース性や長距離依存関係の表現といった課題を抱えていた．

その後，2000 年代後半にはニューラルネットワークを用いたニューラル言語モデルが登場し，単語の意味関係を連続空間上で表現する分散表現（Word Embedding）が導入された．このアプローチにより，語の意味的類似性を数値的に捉えることが可能となり，Word2Vec や GloVe などが広く利用された．さらに，2017 年に Transformer アーキテクチャ [1] が提案され，自己注意機構（Self-Attention）を用いて文中の全単語間の依存関係を効率的に捉えることが可能となった．この革新を契機に，BERT[7] や GPT-2[3] といった事前学習言語モデル（Pre-trained Language Model: PLM）が登場し，大規模データによる事前学習と下流タスクへのファインチューニングという新しい学習規範が確立された．

近年では，この流れをさらに拡張した大規模言語モデルが登場している．膨大なパラメータ数と学習データを活用することで，スケーリング法則（Scaling Law）が成立し，モデルサイズの拡大が性能向上に直結することが示された．GPT-3[4] や PaLM[9], LLaMA[10] などはこの系譜に属し，テキスト生成のみならず，推論・対話・知識検索など多様なタスクにおいて汎用的な能力を発揮している．さらに，ChatGPT の登場以降は，人間の指示に応答できる指示チューニング（Instruction Tuning）や人間

フィードバック強化学習（Reinforcement Learning with Human Feedback: RLHF）といった適応学習手法の発展により，より自然で安全な対話型システムが実現.

言語モデルの発展過程を 4 つの世代に整理すると図 2.1 の通りである．また，近年におけるパラメータ数 100 億以上の大規模言語モデルのタイムラインを図 2.2 に示す．以降，代表的な LLM モデルの詳細を述べる.

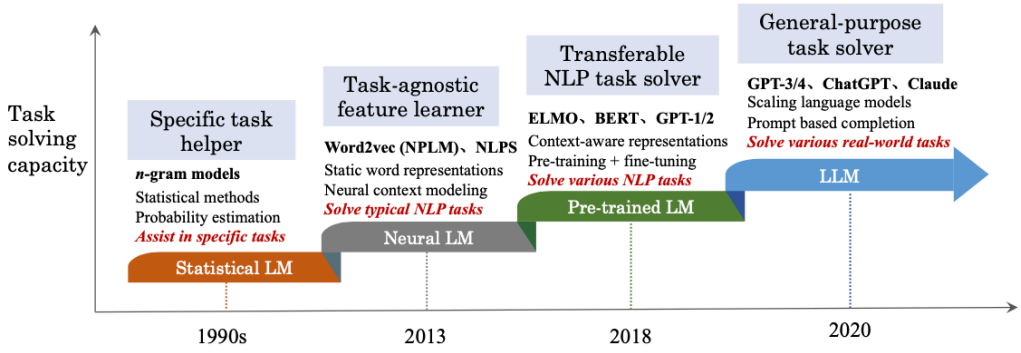

図 2.1: 4 世代の言語モデルの進化過程 [1]

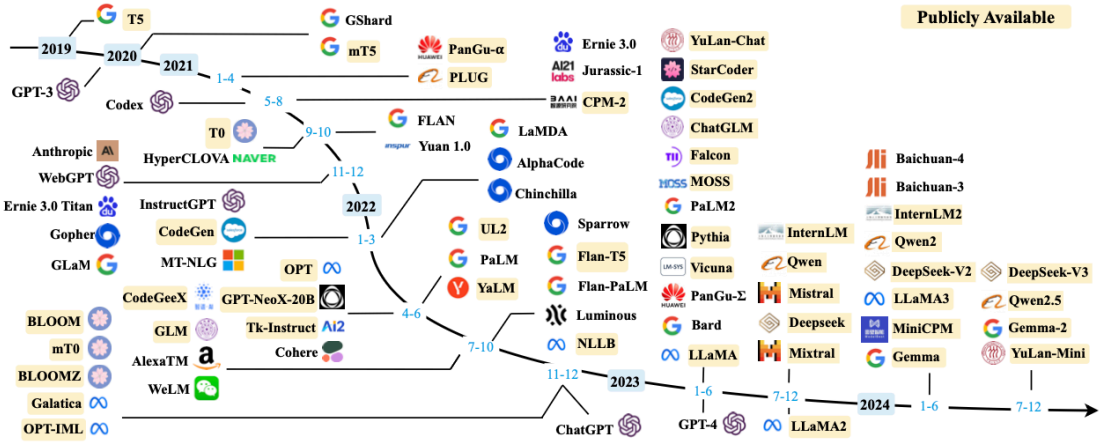

図 2.2: パラメータ数が 100 億以上の LLM のタイムライン [2]

[1] [8] から引用
[2] [8] から引用

2.1.2 Transformer

Transformer は 2017 年に Vaswani ら [1] によって提案された，機械翻訳を目的とするニューラルネットワークモデルである．本モデルは，従来のリカレントニューラルネットワーク（RNN）や畳み込みニューラルネットワーク（CNN）を使用せず，Attention 機構のみを基盤とした構造を持つ点に特徴がある．RNN のような逐次的処理を行わないため，計算の並列化が可能であり，長距離依存関係の学習効率も高い．その結果，Transformer は従来のモデルを上回る翻訳性能を示し，現在では多くの言語モデル（BERT，GPT など）の基本構造として広く利用されている．

アーキテクチャ概要

Transformer は図 2.3 に示すように，エンコーダ・デコーダ型アーキテクチャを採用している．左側のエンコーダが入力文を連続的なベクトル表現へ変換し，右側のデコーダがその情報をもとに出力文を生成する．エンコーダおよびデコーダはそれぞれ 6 層から構成され，各層には Multi-Head Attention 層と Position-wise Feed-Forward Network（FFN）の 2 つのサブレイヤが含まれる．また，それぞれのサブレイヤには Residual Connection と Layer Normalization が組み込まれており，勾配消失の抑制と学習の安定化を実現している．

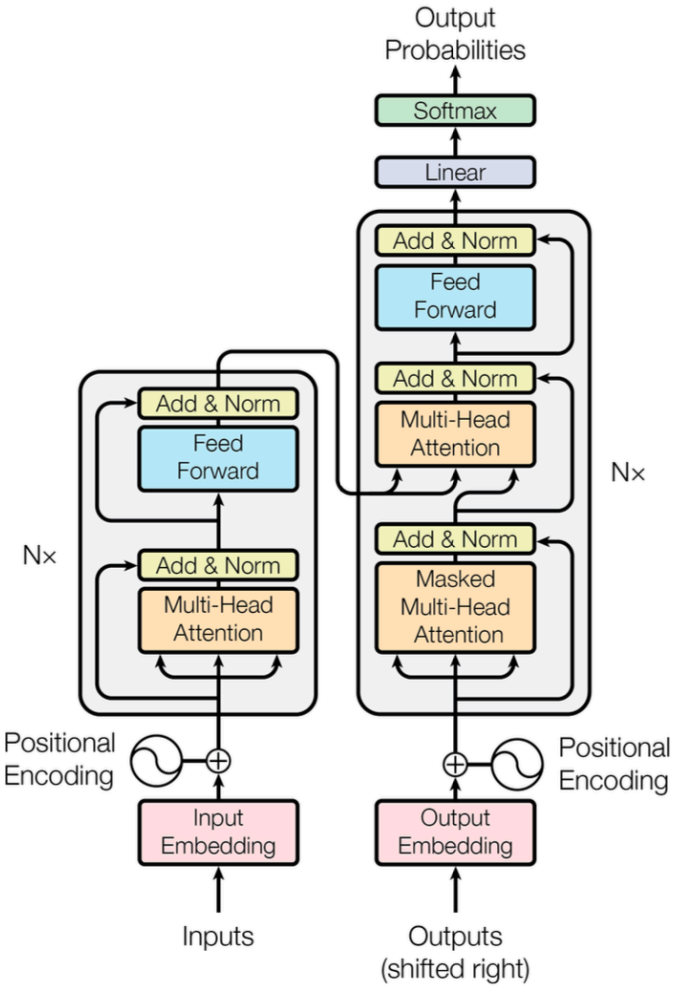

図 2.3: Transformer アーキテクチャの構成 [3]

エンコーダ

エンコーダは，入力系列の各トークン間の関係を学習し，文全体の意味を内包する特徴表現を生成する役割を担う．Self-Attention によって，各トークンは系列中の他の全てのトークンに同時に注意を向けることができ，位置に依存しない文脈依存関係を効果的に捉えることが可能になっている．

[3] [1] から引用

デコーダ

デコーダはエンコーダと似た構造を持つが，出力系列を逐次的に生成するための追加機構を備える．Masked Self-Attention により，未来の単語にアクセスできないよう制限を設け，すでに生成されたトークンのみを参照して次の単語を予測する．また，Encoder-Decoder Attention を通じてエンコーダの出力に基づいた情報を取り込み，入力文と出力文の対応関係を学習する．

Attention 機構

Transformer の中核をなすのが Attention 機構であり，特にその中でも Scaled Dot-Product Attention と Multi-Head Attentnion が主要な要素である．図 2.4 にその構造を示す，左側が Scaled Dot-Product Attention，右側がそれを拡張した Multi-Head Attentnion である．

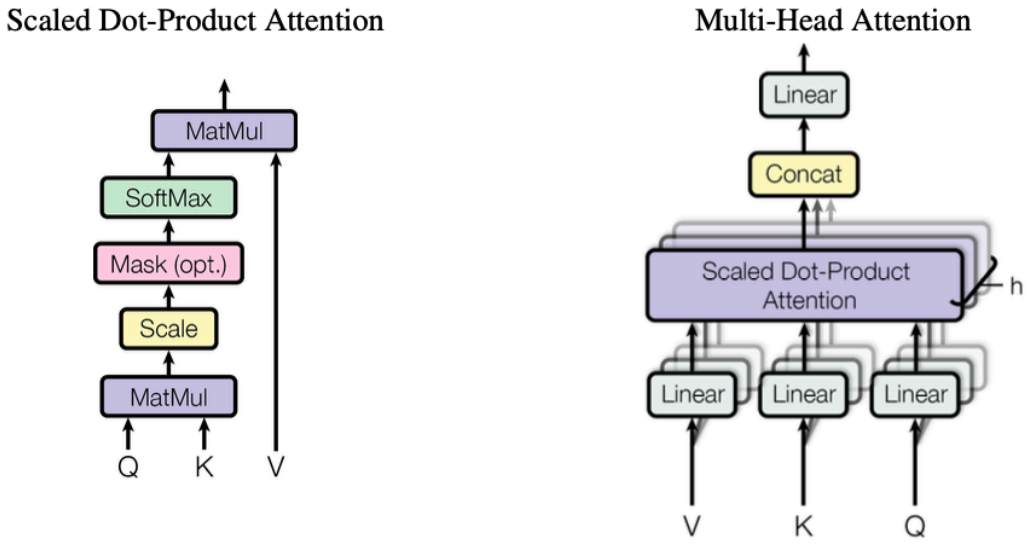

図 2.4: Scaled Dot-Product Attention と Multi-Head Attention の構造 [4]

[4] [1] から引用

Scaled Dot-Product Attention は，入力系列中の単語同士の関連性を定量的に評価し，どの単語にどの程度注意を向けるべきかを計算する仕組みである．入力として Query（Q），Key（K），および Value（V）の 3 種類のベクトルを受け取り，次の式 2.1 によって出力を求める．

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{2.1}$$

ここで，$QK^\top$ により各トークン間の類似度（関連度）を算出し，これを Key の次元数 $\sqrt{d_k}$ の平方根でスケーリングすることで，ソフトマックス関数の出力が極端な値にならないよう調整する．得られた重み分布を Value に掛け合わせることで，入力系列全体の文脈情報を反映した出力が得られる．なお，デコーダにおいては未来の単語を参照しないよう，マスク処理（Mask）を適用する場合がある．

一方，Multi-Head Attention は，この Scaled Dot-Product Attention を複数並列に実行することで，異なる文脈的特徴を同時に捉える仕組みである．図 2.4 の右側に示すように，入力ベクトル Q，K，V はそれぞれ複数の線形変換によって異なる表現中間へ射影される．各ヘッドでは独立に Attention 計算を行い，その出力を Concat（結合）して再び線形変換を施す．これにより，モデルは異なる視点から系列内の依存関係を学習できるようになり，文法的・意味的・位置的関係など多様な特徴を同時に表現することが可能となる．

このように，Attention 機構は従来の RNN のような逐次的処理を必要とせず，系列全体の情報を一度に処理できる点に大きな利点がある．特に Multi-Head Attention によって，Transformer は高い並列性と表現力を両立させ，長距離依存関係を効率的に捉えることを実現している．

2.1.3 GPT

GPT（Generative Pre-trained Transformer）は，OpenAI によって開発された自己回帰型の大規模言語モデルであり，Transformer アーキテクチャ[1] を基盤として構築されている．大量のテキストデータを用いた事前学習によって言語知識や文脈理解能力を獲得し，入力文の続きを自然に生成できる点が特徴である．2018 年に初代モデルである GPT[2]（以降，GPT-1 とする）が発表されて以降，GPT-2[3]，GPT-3[4]，GPT-4[5]，そして GPT-4o[6] へと進化を続けており，言語理解・推論・マルチモーダル処理の各分野で飛躍的な進歩を遂げている．

GPT-2

2019 年に登場した GPT-2 は，GPT-1 の構造を拡張したモデルであり，パラメータ数を 1 億 1700 万から最大 15 億に拡大した．学習には約 40GB に及ぶ WebText データセットを用いており，従来のようなタスクごとの追加学習を行わなくても多様な自然言語処理タスクに対応できる点が特徴であり，翻訳や要約，質問応答といったタスクで高い性能を示し，大規模事前学習の汎用性を実証した．このゼロショット学習の有効性は，以降の大規模言語モデル開発に大きな影響を与えた．

GPT-3

GPT-3 は 2020 年に発表され，パラメータ数を 1750 億にまで拡張した大規模モデルである．GPT-2 と同様の Transformer 構造を基盤としているが，学習データには Common Crawl，Wikipedia，BooksCorpus など膨大で多様なテキストが使用された．GPT-3 の最大の特徴は「Few-shot learning」であり，わずかな例示（プロンプト）を与えるだけで，翻訳・要約・文法補正などの多様なタスクを高精度に実行できる点である．これにより，事前学習済みモデルをファインチューニングせずに多目的に利用できることが示され，言語モデルの柔軟なタスク適応性を確立した．

GPT-4

2023 年に公開された GPT-4 は，GPT-3 の発展版として設計され，テキスト入力だけでなく画像入力にも対応するマルチモーダルモデルである．これにより，画像をもとに説明文を生成したり，図表の内容を解析したりすることが可能になった．性能面では，法律試験（Uniform Bar Exam）で上位 10 %、SAT で 90 %以上のスコアを記録するなど，専門的な試験でも人間に匹敵する水準を達成している．さらに，英語以外の言語における性能向上や，論理的推論能力の強化，生成内容の事実性向上なども報告されている．特にハルシネーションの抑制や安全性の強化に注力した点が特徴であり，信頼性の高い AI モデルとして社会的応用が広がっている．

GPT-4o

2024 年に発表された GPT-4o は，GPT シリーズの中でも最も包括的なマルチモーダル AI モデルである．テキストや画像に加えて，音声や動画も入力として処理し，更には音声での応答も可能となった．これにより，人間との自然な対話が実現し，リアルタイム性の高いインタラクションが可能になっている．また，非英語圏でのタスク性能も向上し，医療・教育・研究など多様な領域での応用が進んでいる．加えて，GPT-4 に比べて，処理速度とコストの両面で効率化が図られており，より軽量かつ実用的なモデルとして位置付けられている．

一方で，人間に近い音声表現や対話能力を持つことで，ユーザーが AI に対して過剰な信頼や感情的な依存を抱く可能性も指摘されている．それでもなお，GPT-4o はマルチモーダル統合と応答品質の両面で大きな進化を遂げたモデルであり，次世代の知的支援システムの基盤として期待されている．

本研究では，視覚的な情報から物体名や関係性を抽出する際に，GPT-4o のマルチモーダル能力を活用することで，オープンボキャブラリかつ高精度なオブジェク

ト認識を実現することを目的とした.

2.2 シーングラフ生成モデル

本節では, シーングラフ生成モデルの歴史と代表的な手法について述べる.

シーングラフ生成（Scene Graph Generation: SGG）は, 与えられた画像からオブジェクトを検出し, それらのオブジェクト間の関係を予測することを目的とするタスクである. 各オブジェクトをノード, 関係をエッジとして表すことで, 画像の内容を構造的かつ解釈可能な形で記述できる点が特徴である. シーングラフ表現は, 画像キャプション生成や Visual Question Answering（VQA）, 画像検索などの様々な下流タスクにおいて有用であることが示されている.

2.2.1 シーングラフ生成の歴史

シーングラフ生成は, 画像内のオブジェクト間の意味的関係を構造的に表現することを目的としたタスクであり, その起源は 2010 年代中頃に登場した視覚的関係認識（Visual Relationship Detection）にさかのぼる. Lu ら [11] は〈主語, 関係, 目的語〉の三つ組で画像を表現する手法を提案し, 物体間の関係理解という新たな方向性を示した. 2017 年には Xu ら [12] によって「Scene Graph Generation」という名称が初めて明確に定義され, 同時期に Visual Genome データセットの公開によって学習環境が整備された。その後, Zellers ら [13] の MotifNet や Yang ら [14] の Graph R-CNN など, 文脈情報やグラフ構造を活用するモデルが登場し, SGG の性能が大きく向上した.

2019 年以降は, オブジェクト間の階層的依存関係を考慮する VCTree[15] などが提案され, 関係推定の精度向上が進んだ. さらに 2021 年以降, Transformer 構造 [1] を導入したモデルが主流となり, SGTR[16] や RelTR[17] など, エンドツーエンドでオブジェクト検出と関係推定を同時に行う手法が登場した。その中でも EGTR（Extracting

Graph from Transformer）[18] は, オブジェクトと関係を統一的にトークンとして扱うことで高い表現力を実現した.

近年では, CLIP[19] や Grounding DINO[20] などのマルチモーダル事前学習モデルを活用し、未知のカテゴリや関係にも対応可能なオープンボキャブラリ型 SGG（Open-Vocabulary SGG）への発展が進んでいる.

以下, 代表的な手法を述べる.

2.2.2 EGTR

EGTR[18] は，Transformer ベースの DETR[21] に内在する自己注意（Self-Attention）の重みから，物体間の関係を直接抽出する SGG モデルである．既存の RelTR や SGTR が別途「triplet query」や「triplet detector」を導入していたのに対し，EGTR は DETR の自己注意層におけるクエリ間の依存構造をそのまま関係情報として再利用する．これにより，モデルの軽量化と高速な推論を実現している．

モデル構造

図 2.5 に EGTR の全体構造を示す．EGTR は，DeformableDETR をベースとし，まず CNN および Transformer エンコーダにより画像特徴を抽出し，デコーダの Self-Attention 層においてオブジェクトクエリ間の関係を学習する．デコーダの Self-Attention 層から得られるクエリ（Q^l）とキー（K^l）を主語・目的語とみなして関係表現を構築する：

$$R_a^l = \left[Q^l W_S^l \,;\, K^l W_O^l \right], \tag{2.2}$$

ここで, W_S^l, W_O^l は主語･目的語の射影行列であり，得られたテンソル $R_a^l \in \mathbb{R}^{N \times N \times 2d_{\text{model}}}$ は各物体ペア間の関係特徴を表す．また，最終層のオブジェクト表現 Z^L に対して

も同様の処理を施し，物体検出情報を補完する：

$$R_z = \left[Z^L W_S \, ; \, Z^L W_O \right]. \tag{2.3}$$

これら全層の関係表現を Gated Sum で統合し，情報の寄与度を動的に制御する：

$$\hat{G} = \sigma \left(\mathrm{MLP}_{\mathrm{rel}} \left(\sum_{l=1}^{L} \left(g_a^l \odot R_a^l \right) + g_z \odot R_z \right) \right), \tag{2.4}$$

ここで，g_a^l, g_z はゲート値，$\mathrm{MLP}_{\mathrm{rel}}$ は 3 層の全結合層，σ はシグモイド関数である．この出力 $\hat{G} \in \mathbb{R}^{N \times N \times |C_p|}$ は，各物体ペアの述語（関係）確率を表すシーングラフとなる．

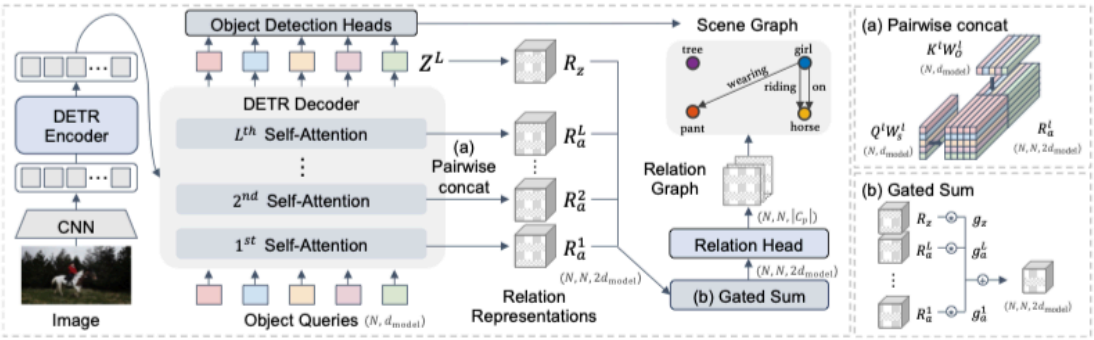

図 2.5: EGTR アーキテクチャの構造 [5]

EGTR は，物体検出と関係抽出を同時に最適化するマルチタスク学習を採用する．総損失は以下で定義される：

$$\mathcal{L} = \mathcal{L}_{\mathrm{od}} + \lambda_{\mathrm{rel}} \mathcal{L}_{\mathrm{rel}} + \lambda_{\mathrm{con}} \mathcal{L}_{\mathrm{con}}, \tag{2.5}$$

ここで，$\mathcal{L}_{\mathrm{od}}$ は物体検出損失，$\mathcal{L}_{\mathrm{rel}}$ は関係抽出損失，$\mathcal{L}_{\mathrm{con}}$ は Connectivity Prediction（2 値分類による関係有無判定）の損失である．

[5] [18] から引用

また，学習初期の検出誤差が関係推定に悪影響を及ぼすことを防ぐため，Relation Smoothing を導入する．物体 i に対する不確実性 u_i をマッチングコストから算出し，

$$u_i = \sigma\big(\mathrm{cost}_i - \mathrm{cost}_{\min} + \sigma^{-1}(\alpha)\big)\,, \tag{2.6}$$

主語・目的語の信頼度に基づき関係ラベルを補正する：

$$G_{ijk} = (1 - u_i)(1 - u_j). \tag{2.7}$$

これにより，学習初期は物体検出を優先し，訓練が進むにつれて関係抽出へと焦点が移るカリキュラムの学習が実現される．

しかし，EGTR は Visual Genome[22] や Open Images[23] などのデータセットによる学習によって一定の性能を示している．つまり，特定のデータセットに含まれるカテゴリラベルに依存しており，学習時に存在しなかった語彙を認識・関係推定することは困難である．

このような制約を克服するため，近年ではオープンボキャブラリシーングラフ生成（Open-vocabulary Scene Graph Generation; OvSGG）が注目されている．これらの手法では，大規模視覚言語モデル（Vision-Language Model; VLM）や大規模言語モデルを活用することで，訓練データに含まれない新規カテゴリや関係語にも一般化することを目指している．

次節では，このオープンボキャブラリシーングラフ生成の代表的な研究について述べる．

2.2.3 オープンボキャブラリシーングラフ生成

オープンボキャブラリシーングラフ生成は，未知の物体や関係語を副務画像に対しても柔軟に対応することを目的としたタスクである．本節では，代表的な研究と

して，OvSGTR[24] と PGSG[25] を取り上げる．

OvSGTR

2023 年に Chen ら [24] は，OvSGTR を提案し，ノード（物体）とエッジ（関係）の両方を対象とした完全オープンボキャブラリ SGG を実現した．本手法は，従来の SGTR[26] を基盤とした Transformer ベースのエンコーダデコーダ構造を採用しており，画像特徴から物体および関係を同時に推定するエンドツーエンドのフレームワークである．

彼らは，シーングラフ生成を表 2.1 に示す通り，4 つの設定に分類し，特に物体と関係の両方が未知となる最も困難な状況に対して統一的な学習フレームワークを構築している．

表 2.1: OvSGTR における 4 つのシーングラフ生成設定の概要

設定名	物体	関係	概要
Closed-set SGG	既知	既知	学習時と同一の物体・関係カテゴリで推論を行う標準的な設定.
OvD-SGG	未知を含む	既知	物体クラスのみをオープン化した設定．未知の物体を検出しつつ，既知の関係語で関係を予測する．
OvR-SGG	既知	未知を含む	関係語のみをオープン化した設定. 既知の物体ペアに対して未知の関係語を推定する．
OvD+R-SGG	未知を含む	未知を含む	物体と関係の両方をオープン化した最も困難な設定．未学習カテゴリを含む物体検出と関係推定を同時に行う．

提案手法では，画像キャプションデータを用いた弱教師学習により，視覚特徴とテキスト概念を整合させる Visual-Concept Alignment を導入し，さらに Knowledge

Distillation に基づく Visual-Concept Retention 戦略によって関係カテゴリの忘却を防いでいる．これにより，VG150 データセットにおいて既存手法を上回る性能を示し，未知の関係を含むシーンにも対応可能であることを示した．

OvSGTR における VLM の活用は，CLIP や Grounding DINO などの判別型モデルを利用して視覚と言語の埋め込み空間を整合させる特徴整合型アプローチであり，VLM はあくまで意味空間の橋渡し役として機能している．なお，LLM による生成的推論は導入されておらず，今後の課題として言及されている．

PGSG

2024 年に Li ら [25] は PGSG を提案し，生成型 VLM（BLIP，InstructBLIP）を用いて画像から直接シーングラフを生成する新たな枠組みを示した．図 2.6 に PGSG の全体構造を示す．

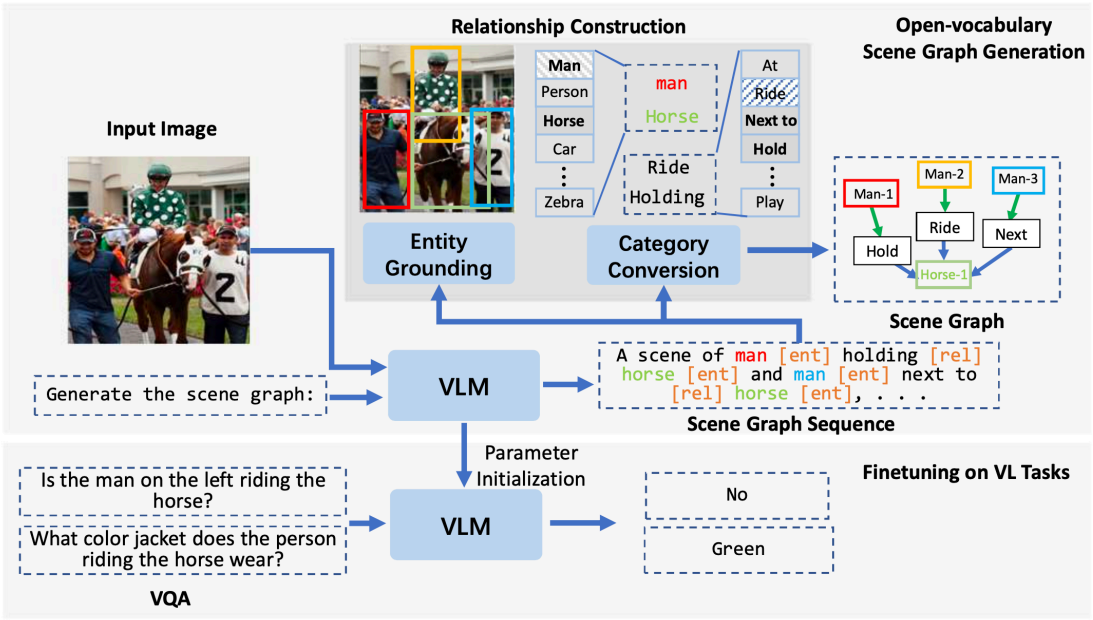

図 2.6: PGSG のパイプライン [6]

[6] [25] から引用

PGSG は，画像からシーングラフを直接生成する image-to-graph generation パラダイムを採用し，画像と言語の理解・生成の統一を行う BLIP[27] や，ユーザの指示に応答可能な InstructBLIP[28] などの生成 VLM を基盤としている．本手法の最大の特徴は，従来のように物体検出と関係分類を別段階で行うのではなく，VLM の言語生成能力を活用して画像をテキスト列（scene-graph sequence）へと変換する点にある．

まず，入力画像は VLM の視覚エンコーダにより特徴量へと変換され，テキストデコーダがそれに基づき，「man [ent] holding [rel] horse [ent]」のようなシーングラフ文を生成する．この出力文では，[ent] トークンが物体，[rel] トークンが関係を表しており，自然言語の構文の中に関係構造を埋め込むことで，オープン語彙の関係表現を自然に生成できる．

次に，生成された文は Relationship Construction モジュールに入力され，ここで 2 段階の処理が行われる．

Relationship Construction モジュール

1. **Entity Grounding**

   生成文に含まれる各エンティティに対応する画像領域を推定し，バウンディングボックスを決定する．これにより，文中の「man」や「horse」と画像中の実際の物体が対応付けられる．

2. **Category Conversion**

   生成された語彙をデータセット固有のクラス空間にマッピングし，標準的なシーングラフ形式（ノードとエッジの集合）に変換する．

このような関係構築を経て，図 2.6 の右上に示すようなシーングラフが得られる．PGSG はこの一連のプロセスを通じて，既知・未知を問わず多様な関係の表現を可能にしたオープンボキャブラリ SGG を実現している．

さらに，生成過程で得られたVLMのパラメータは他の視覚言語タスクにも転移可能である．例えば，VQAにおいて「左側の人物は馬に乗っているか」や「馬に乗っている人物のジャケットの色は何か」といった質問に対して，シーングラフ生成で学習した関係表現を活用して回答を行うことができる．このようにPGSGはシーングラフ生成と視覚言語タスクの統一的フレームワークを実現しており，VLMの生成的能力をシーン理解へと拡張している．

このように，オープンボキャブラリシーングラフ生成の研究は，VLMを特徴整合のための基盤モデルとして活用する段階から，生成を通じてシーン理解を拡張する段階へと発展しつつある．VLMの役割が「意味空間の共有」から「言語的生成による知識表現」へと広がる中で，シーングラフ生成の枠組みもより柔軟で表現的な方向へと進化している．

本研究では，

2.3 知識グラフ

2.3.1 知識グラフとは

Lisa Ehrlingerらによれば，知識グラフとは「情報をオントロジーに統合し，推論エンジンを用いて新しい知識を導出するものである」と定義されている．また，Hoganらは「現実世界の知識を蓄積し，それを伝達することを目的としたデータのグラフ」であると述べている．知識グラフは，現実世界の実体（エンティティ）をノードとして，それらの間の関係（リレーション）をエッジとして表現する構造を持つ．このような構造により，単なるデータの集積ではなく，エンティティ間の意味的つながりを明示的に記述できる点が特徴である．

知識グラフのエッジは，たとえば???? といったように，意味のある関係を示す．このため，知識グラフは構造的な情報を活用して，検索や質問応答，推論などの高度

なタスクに応用できる．また，知識グラフには様々な表現形式が存在し，有向エッジラベル付きグラフ（directed labeled graph），ヘテログラフ（heterogeneous graph），プロパティグラフ（property graph）などの形式が用いられる．

近年では，Google の Knowledge Graph に代表されるように，Web 上の膨大な情報を整理・統合する基盤としても利用されている．このように，知識グラフは情報を意味的に関連づけ，機械が理解・推論可能な形で知識を表現する枠組みとして重要な役割を果たしている．

第3章　提案手法

第 3 章では，LLM を用いたシーングラフ生成の手法について述べる．

3.1　使用モデル

3.1.1　Grounding DINO

Grounding DINO[20] は，Liu らによって提案されたオープンセット物体検出モデルである．従来の物体検出モデルは COCO などのデータセットに含まれる固定カテゴリしか検出できないという制約があった．これに対して，Transformer ベースの物体検出器である DINO を基盤とし言語情報を導入することで，自然言語の入力を条件として任意の物体を検出できるように設計されている．そのため，未知のカテゴリを含むオープンワールド環境でも柔軟に対応可能である．

Grounding DINO は，画像とテキストの情報を密接に統合するために，Dual-encoder-single-decoder アーキテクチャを採用している．従来モデルが一部の段階でしか特徴量を融合していなかったのに対し，Grounding DINO では以下の 3 つのフェーズで視覚と言語を融合させている．図 3.1 に Grounding DINO のフレームワークを示す．

Dual-encoder-single-decoder アーキテクチャ

1. Feature Enhancer
   自己注意機構に加え，テキスト画像，画像からテキストへの双方向の交差注意機構（Cross-Attention）を積み重ねることで，特徴量を強化・融合する．

2. Language-guided Query Selection
   入力テキストに最も関連性の高い画像特徴をデコーダのクエリとして選択する．

3. Cross-modality Decoder
   画像とテキストの両方の特徴量をクエリに注入し，モーダル間のアライメントをさらに高め，最終的な物体ボックスとラベル語句を出力する．

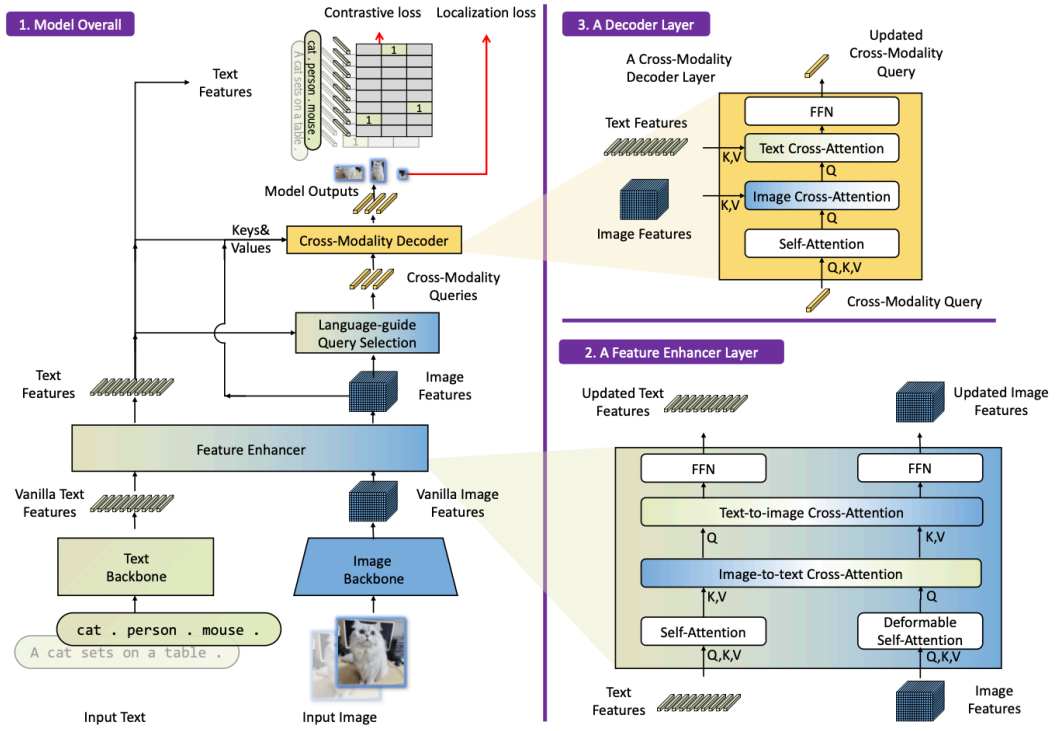

図 3.1: Grounding DINO のフレームワーク [1]

さらに，複数のカテゴリ名を入力する際，無関係な単語同士が干渉し合うのを防ぐため，Sub-sentence level Representation という技術を導入している．Attention

[1] [20] から引用

mask を用いて，特定の単語間の関連性を強制的に遮断し，単語ごとの細かい情報を保持している．学習には，物体検出データ（Objects365，OpenImages，COCO），グラウンディングデータ（Flickr30k，Visual Genome，RefCOCO など），キャンプションデータ（Cap4M）を含む大規模データセットで事前学習を行うことで，高度な概念の一般化を実現している．

3.1.2 SAM

3.2 シーングラフ生成手法

本節では LLM を用いたシーングラフ生成手法について述べる．本手法の一連の流れは図の通りである．

3.2.1 物体列挙

このステップでは，画像を LLM に入力し，画像内に含まれる可能性のある主要な物体を列挙する．これは未知の物体名を含むオープンボキャブラリな物体候補集合を得ることを目的とする．出力は修飾語を含まない単純名詞のリストであり，以下のプロンプトを LLM（GPT-4o）に与えることで行う．なお，実際のプロンプトは付録に示す．

プロンプトの概要

以下の指示，出力形式に基づいて，画像内の物体を抽出することを指示
#指示

1. 画像内に見える**主要な物体**を全て列挙
2. 各要素は**単純な単数系の名詞**
3. **形容詞・色・素材・状態**などは含めない
4. **認識された物体名のみ**を出力する

#出力形式
カンマで区切る

大規模言語モデルは，事前学習によって膨大な知識や文脈理解能力を獲得しているが，その能力を効果的に引き出すためにはプロンプトの設計が極めて重要である．プロンプトはモデルに対する入力文であり，その内容や構造によって出力結果の精度・一貫性・創造性が大きく変化する．そこで上記のプロンプトでは，出力形式を明確に定義することでノイズを排除し，語彙の統一性を保つことで後段の処理を容易にすることを目的として設計した．このような設計により，モデルが曖昧な表現や冗長な記述を避け，画像内の実体に対応する一般的な語彙を抽出できるようになる．

なお，ここで得られた物体名の集合を

$$\hat{O} = \{\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_n\} \tag{3.1}$$

とする．

3.2.2 座標取得

ここでは得られた物体名をテキストクエリとして Grounding DINO（第 3.1.1 項）に入力し，対応するバウンディングボックス $box = (x_1, y_1, x_2, y_2)$ と信頼度を得る．ここで，物体ごとの中心座標は次式で求める．

$$\mathbf{c}_i = \left(\frac{x_1 + x_2}{2},\ \frac{y_1 + y_2}{2} \right) \tag{3.2}$$

さらに，各物体領域をより明確に抽出するため，SAM（第 3.1.2 項）によるマスク生成を併用する．SAM は，Grounding DINO で得られた矩形領域を入力とし，物体境界をピクセルレベルで抽出する．これにより，矩形境界のみでは不明確だった物体の輪郭が精密に得られ，重なり合う物体や接触判定における曖昧性を低減することができる．SAM の出力は，COCO 形式のポリゴン表現 $segmentation = (x_0, y_0, x_1, y_1, \ldots, x_n, y_n)$ で得る．これにより，物体 o_i の位置情報として，矩形＋ポリゴンのハイブリッド表現が得られる．

検出された物体を $\mathcal{O} = \{o_1, o_2, \ldots, o_N\}$ とし，画像に存在しない物体（Grounding DINO で検出されなかった候補）は自動的に除外される．本処理により，LLM が列挙した候補のうち実際に画像に存在するものだけが残るため，フィルタリングの役割を果たす.

3.2.3 関係語生成

本ステップでは，第 3.2 項で得られた物体集合 $\hat{O} = \{\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_n\}$ に対して，全ての物体ペア (o_i, o_j) を組み合わせ，それぞれの関係を大規模言語モデルによって生成する．生成結果は，関係語（relation）に加え，位置関係（position）および接触関係（attach）を含む JSON 形式で出力される．

関係語の生成には，LLM に明確な出力仕様を与えるため，指示文と出力形式を明示した以下のプロンプトを使用する．このプロンプトは，抽象的な意味関係ではなく，日常的・物理的に成立する空間関係を想定して設計している．また，プロンプト内に具体例な出力例を含めることで，LLM の出力を安定化させ，より一貫した関係文の生成を促している．なお，実際に使用したプロンプト全文は付録に示す.

プロンプトの概要

現実世界の日常的な空間関係や物体の配置に焦点を当て，抽象的で孤立した記述ではなく，現実で物体的にもっともらしい関係の生成を指示

#指示

1. n 個の文章を生成すること
2. 各項目は以下のキーを持つ JSON 形式で出力すること
   - ”sentence”：現実的な空間関係を明確に表す文脈に基づいた文章
   - ”object”：比較対象のオブジェクトのリスト
   - ”relation”：表現される空間的関係
   - ”position”：3 次元空間でどの位置にあるか
   - ”attach”：物体が物理的に接触しているか
3. 出力全体を JSON 配列で包む
4. コメントや説明を含めない

このプロンプト設計により，モデル出力をそのまま機械処理に利用できるようになっている．以下に，生成方法の詳細を解説する．

2 物体を含む文の生成

まず，列挙された物体集合 $\mathcal{O}$ の全ての組み合わせ (o_i, o_j) に対して，LLM の持つ一般的な知識を活用し,「現実世界で起こりうる」関係を文形式で大量に生成する．

ここで，文の主語は必ずしも固定されず，どちらの物体を主語としてよいものとする．これにより,「机の上にりんごがある」「りんごの下に机がある」など，同一の関係を異なる視点から柔軟に表現できる．

また，単純に 2 物体を含む文を生成させるだけでは,「There is an apple on the table.」のような幾何的な記述が中心となり，日常的なシーンを反映した文が得られにくい．そこで，LLM に対し**文脈的に自然な状況を反映するヒント（contextual**

hints）を含めるよう指示することで,「The book is on the desk in an office.」や「The mug is placed next to the sink in the kitchen.」のように，現実的なシーンとして成立する文の生成を誘導する．この設計により，得られる関係データは単なる座標的関係にとどまらず，意味的に妥当な日常シーンの中で成立する関係表現も獲得される．

生成された文は”sentence”フィールドに格納し，対応する物体ペアは”object ”にリスト形式で記録する．さらに，この情報をもとに”relation”に関係語を格納する．

位置関係の生成

関係語の生成をする際，単に 2 物体を含む文章を生成させるだけでは，後段の探索過程で得られる関係推論が曖昧になりやすい．例えば,「apple」と「table」の 2 つの物体が水平方向に位置し，正しい関係は「next to」である場合でも,「on」という関係が一般的な文脈で頻出するため，物体名のみを入力として探索を行うと「on」が誤って選択されてしまう可能性がある．つまり，それらがどのような位置情報にあるかを明示されなければ，グラフ探索時に両者を結ぶ経路を適当に特定することができず，結果として推論が偶発的で不安定なものとなってしまう．

そこで本研究では，画像から得られる位置情報（第 3.2.2 項）を探索の手がかりとするため，LLM に対して文章生成と同時に 2 物体間の空間的な位置関係を明記させるように設計する．これにより,「apple」と「table」のような語彙的関係にとどまらず，空間的に整合する構造的情報を持つようになる．

具体的には，各文において Object1 が Object2 に対して，垂直方向（vertical），水平方向（horizontal），奥行方向（depth）の 3 軸における相対的な位置を，up，down，left，right，front，back，None のいずれかで符号化する．

$$\text{vertical} \in \{\text{up}, \text{down}, \text{None}\},\ \text{horizontal} \in \{\text{left}, \text{right}, \text{None}\},\ \text{depth} \in \{\text{front}, \text{back}, \text{None}\}$$

例えば「The apple is on the table.」の場合,「apple」は「table」の上方向に存在するため,

$$\text{position} = \{\text{vertical} : \text{down},\ \text{horizontal} : \text{None},\ \text{depth} : \text{None}\}$$

として出力される.

このようにして，関係語に加え位置情報を保持することで，意味的関係と空間的配置の両面を反映したネットワーク構造を構築できる．後続のグラフ探索において，単語の出現頻度に依存しない空間的に妥当な重み付けに基づく関係推論が可能になる.

接触関係の生成

さらに，物体間の関係をより正確に表現するため，物理的な接触の有無を明示的に生成させる．位置情報のみでは関係性を十分に区別できない場合が多く，例えば「apple」が「table」の上に位置している場合，その関係は「on」である可能性もあれば「above」である場合もある．しかし，位置情報だけでは両者の違いを識別できず，いずれも垂直方向（vertical）が”up”であるという同一の特徴として扱われてしまう．その結果，本来「above」である関係が「on」として出力されるなど，関係語が誤って推定される可能性がある.

この問題を解消するため，本研究では LLM に対して，生成された文が想定する状況において 2 物体が接触しているか否かを同時に出力させる．接触情報は”attach”に記録され，その値は”True”または”False”のいずれかで表される．基準は以下の通りである.

接触判定

- ”True”：物体が実際に接している
  （例：placed on, touching, leaning, stacked, held など）
- ”False”：空間的に近くても直接の接触はない
  （例：in front of, next to, above など）

このように接触情報を明示的に付与することで,「垂直方向に上方にある」という単純な位置情報だけでなく，物理的な相互関係を含む関係性を捉えることが可能となる．得られた”attach”フィールドは，後段の TripletRanker において探索区間の制約条件として用いられ，物理的に矛盾しない三つ組〈主語, 関係, 目的語〉の推論を助ける役割を果たす.

エラー処理

LLM による出力は確率分布に基づく生成過程を通じて行われるため，同一のプロンプトを与えても常に完全に同一の結果が得られるとは限らない．出力形式や構造を詳細に指定した場合においても，トークン生成時の確率的揺らぎや文脈解釈の不確実性により，意図しない形式の文章や構造上の誤り，あるいは値の欠落を生じることがある，これは，LLM が厳密な規則に従って動作するルールベースモデルではなく，膨大な事例から統計的に言語パターンを学習した確率的生成モデルであることに起因する．そこで本研究では，このような生成結果の不確実性に対処するため，出力が事前に定義したフォーマットに適合しているかを自動的に検証する機構を導入する。具体的には、生成結果において必要な値が欠落している場合や、構造がフォーマット仕様と一致しない場合を検出し、それらが確認された際には再生成を自動的に実行する．これにより，生成結果の構造的信頼性を確保しつつ、プロンプトの自由度と創発性を維持したまま、安定した関係データの取得を実現している。

3.2.4 関係グラフ構築

本ステップでは，前段で得られた関係データをもとに，物体，関係語，位置関係，および接触情報を統合して関係グラフを構築する．構築される関係グラフは，有向グラフ $G = (V, E)$ で表され，ノード集合 V は物体や関係に関する要素を，エッジ集合 E はそれらの共起確率を表す．各ノードは，画像中の物体と LLM 出力の意味

情報を対応づけ，関係文から抽出された主語・関係語・目的語を多段的に結ぶ構造を持つ．これにより，単なる文表現のつながりではなく，空間的配置と物理的生合成を伴う関係ネットワークを表現できる．構築する関係グラフの一例を図に示す．

（図）

ノード設計

本研究で用いるノード種別を表 3.1 に示す．これらのノードは，物体 → 位置付き物体 → 関係表現 → 三つ組という階層的接続を通じて結ばれており，言語的関係と空間的配置を統合的に扱うことが可能となっている．

表 3.1: ノード種類ごとの例と役割

ノード種別	例	役割
object	apple	画像内物体（主語・目的語）を表す基本ノード
area_object	up_apple	主語側の位置情報を付与したノード（up/down/left/right/front/back）
relation_object	apple on	主語と関係語を結合したノード
triplet	apple on table	完全な三つ組（主語・関係語・目的語）
relation_object_to	on table	関係語と目的語を結合したノード
area_object_to	down_table	目的語側に反転位置を付与したノード（逆方向探索用）

area_object ノードは，言語的な関係表現が必ずしも一意な物理配置を示さないという問題を解決するために導入する．例えば,「The apple is on the table.」という文を考えると，人間は直感的に「リンゴが机の上にあり，上方向の接触が生じている」状況を理解できる．しかし同じ”on”には,「壁にかかった絵」や「ドアに貼られたメ

モ」のように，接触方向が上方向ではなく側面方向である場合にも用いられる．このとき，”on”を単独の関係語として集約し，単純に object ノード →relation_object ノード（apple→apple on）のような表現だけでグラフを作ると，異なる物理配置に基づく”on”が同一ノードに混在してしまう．その結果，探索時に参照される重みは「”up”だから”on”になった」「”in front of”だから”on”になった」といった位置条件を反映したものではなく,「単に”on”という単語がよく生成されたから」という頻度バイアスに引っ張られた結果になり得る．これでは，探索経路が表す意味が曖昧になり，なぜその関係を選んだのかを位置情報に基づいて説明できない．

そこで本研究では，位置情報をノードとして明示化し，up_apple のような area_object ノードを relation_object ノードの前段に挿入する．この設計により,「上方向にある”apple”からは”on”/”above”が候補になりやすい」といった位置条件付きの関係傾向がグラフ上に分離され，探索が単語頻度に支配されることを抑制できる．

さらに，目的語側にも area_object_to ノードを生成する．これは記述された位置情報の逆方向（up↔down，right↔left，front↔back,）に基づいて，主語側の位置属性と整合する形で目的語側の位置を表現するものである．これにより，探索は主語側だけに依存せず，目的語側からも一貫した制約で関係を評価できる．

エッジ設計と重み付け

接触関係の付与

3.2.5 グラフ探索

3.2.6 シーングラフ生成

第4章　評価・実験

第5章　結論

ここに結論

謝辞

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need.

[2] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners.

[5] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report, 2023.

[6] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding.

[8] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models.

[9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways.

[10] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models.

[11] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors.

[12] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing.

[13] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context.

[14] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-CNN for scene graph generation.

[15] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts.

[16] Rongjie Li, Songyang Zhang, and Xuming He. SGTR: End-to-end scene graph generation with transformer.

[17] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. RelTR: Relation transformer for scene graph generation.

[18] Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. EGTR: Extracting graph from transformer for scene graph generation.

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[20] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024.

[21] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016.

[23] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, Vol. 128, No. 7, p. 1956–1981, March 2020.

[24] Zuyao Chen, Jinlin Wu, Zhen Lei, Zhaoxiang Zhang, and Changwen Chen. Expanding scene graph boundaries: Fully open-vocabulary scene graph generation via visual-concept alignment and retention, 2024.

[25] Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models, 2024.

[26] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer, 2022.

[27] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[28] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.