第1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 1.1 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 1.2 節ではシーングラフ生成に関連する研究を取り上げる．

1.1　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 Procesing: 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, インターネット上の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[?] 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 mechanism）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 GTP シリーズや Google の BERT 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.

1.1.1 LLM の歴史

Zhao らのサーベイ論文 [?] 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[?] 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[?] や GPT-2[?] 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（Large Language Model: LLM）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[?] や PaLM[?]，LLaMA[?] 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 1.1 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 1.2 に示す．以降，代表的な LLM モデルの詳細を述べる．

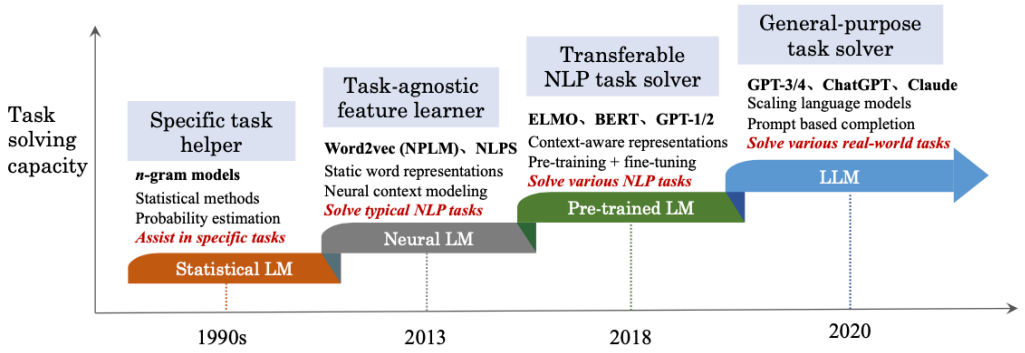

図 1.1: 4 世代の言語モデルの進化過程 [1]

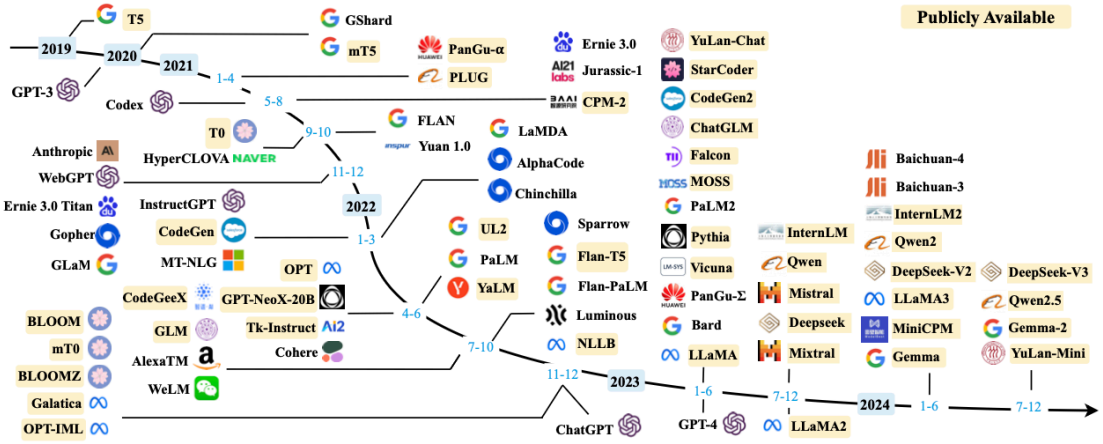

図 1.2: パラメータ数が 100 億以上の LLM のタイムライン

1.1.2 Transformer

Transformer は 2017 年に Vaswani ら [?] 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

[1] [?] から引用
[1] [?] から引用

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 1.3 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 層から校正され，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残差接続（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．

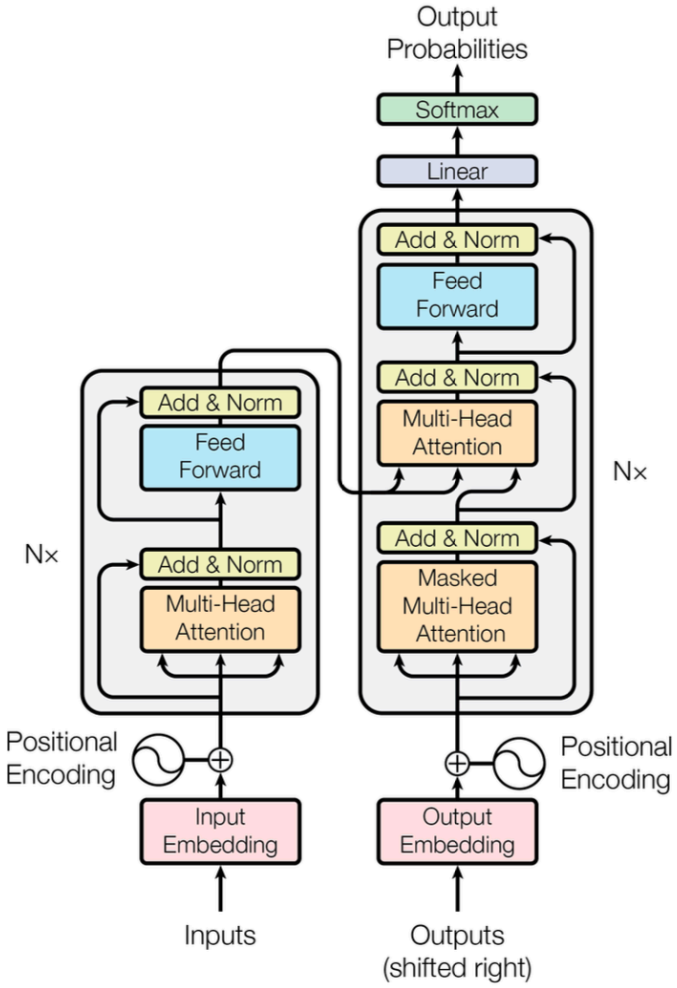

図 1.3: Transformer アーキテクチャの構成

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．

[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 1.4 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．

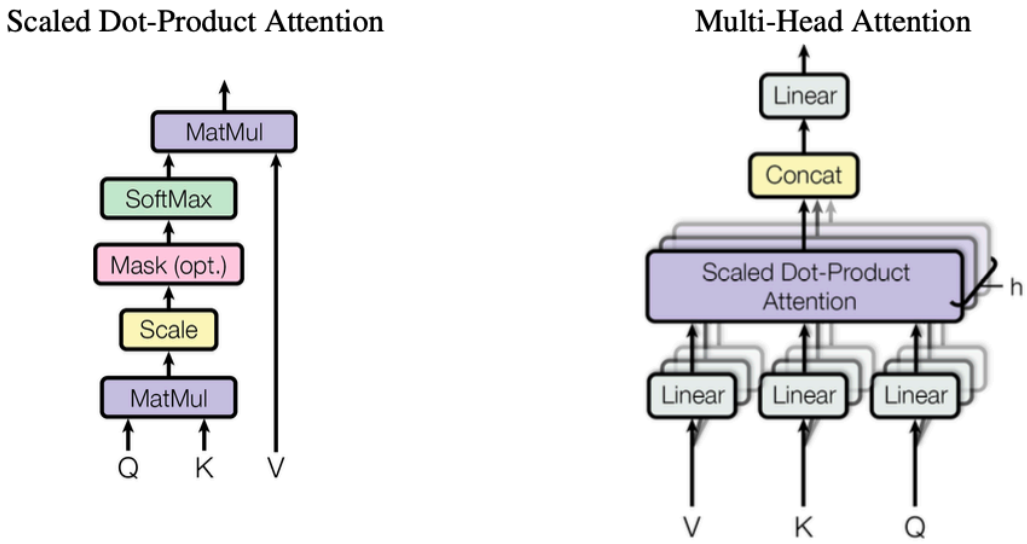

図 1.4: Scaled Dot-Product Attention と Multi-Head Attention の構造

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

[1] [?] から引用

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 1.1 によって出力を求める．

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1.1}$$

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 1.4 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．

1.1.3 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[?] 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-1[?] が発表されて以降，GPT-2[?]，GPT-3[?]，GPT-4[?]，そして GPT-4o[?] 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る．特定タスクに特化したファインチューニングを行わずとも，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 億にまで拡張した超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．

1.2　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.

1.2.1 シーングラフ生成の歴史

シーングラフ生成（Scene Graph Generation: SGG）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 [?] 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 [?] 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 [?] の MotifNet や Yang ら [?] 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[?] 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 [?] を導入したモデルが主流となり, SGTR[?] や RelTR[?] 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（End-to-End Graph Transformer）[?] 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 や Grounding DINO 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.

1.2.2 EGTR

EGTR[?] は，Transformer ベースの 1 段階物体検出器（DETR）[?] 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 1 段階 SGG モデル（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 1.5 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：

$$R_a^l = \left[Q^l W_S^l \,;\, K^l W_O^l \right], \tag{1.2}$$

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：

$$R_z = \left[Z^L W_S \,;\, Z^L W_O \right]. \tag{1.3}$$

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：

$$\hat{G} = \sigma \left(\text{MLP}_{\text{rel}} \left(\sum_{l=1}^{L} \left(g_a^l \odot R_a^l \right) + g_z \odot R_z \right) \right), \tag{1.4}$$

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．

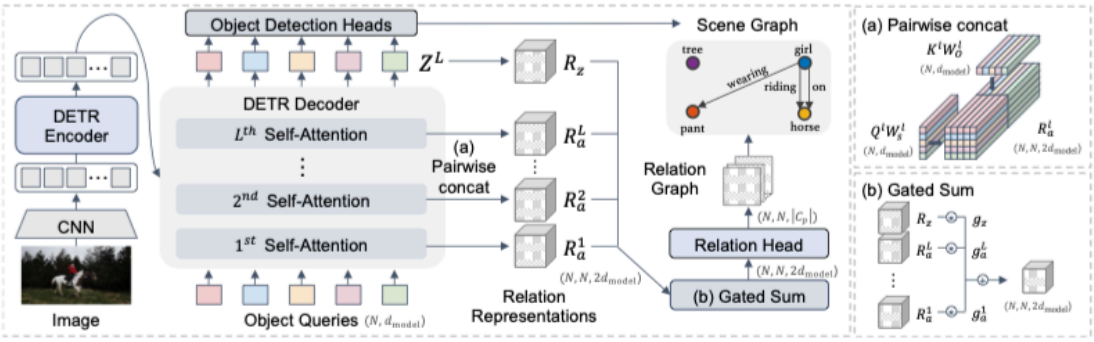

図 1.5: EGTR アーキテクチャの構造

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：

$$\mathcal{L} = \mathcal{L}_{\mathrm{od}} + \lambda_{\mathrm{rel}}\mathcal{L}_{\mathrm{rel}} + \lambda_{\mathrm{con}}\mathcal{L}_{\mathrm{con}}, \tag{1.5}$$

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．

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，

$$u_i = \sigma\big(\mathrm{cost}_i - \mathrm{cost}_{\min} + \sigma^{-1}(\alpha)\big)\,, \tag{1.6}$$

主語・目的語の信頼度に基づき関係ラベルを補正する：

$$G_{ijk} = (1-u_i)(1-u_j). \tag{1.7}$$

[1] [?] から引用

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[?] や Open Images 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，未知の物体や関係を扱うことはできない．すなわち，EGTR はクローズドボキャブラリ設定で設計されており，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（LLM）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．

1.2.3　オープンボキャブラリシーングラフ生成