令和 4 年度茨城大学大学院理工学研究科情報工学専攻

修士学位論文

CLIP を用いた Visual WSD

所属　情報工学専攻

著者　伊藤陽樹 (21NM708A)

指導教員　新納浩幸教授

令和 5 年 2 月 3 日 (金)

令和 4 年度茨城大学大学院理工学研究科情報工学専攻　修士学位論文

CLIP を用いた Visual WSD

著者

伊藤陽樹 (21NM708A)

指導教員

新納浩幸教授

論文要旨

近年，画像もしくは動画を対象とした研究と自然言語を対象とした Vision-and-Language という研究が，大きな広がりを見せている．CLIP は，画像と言語の 2 つの領域の埋め込み表現を得ることができる事前学習モデルの一つである．CLIP のような事前学習モデルは，大規模コーパスで事前学習し，zero-shot で様々な V&A 分野のタスクを高精度に処理することができる．CLIP は，CNN よりも効率性の良い VisionTransformer を画像エンコーダに利用している．CLIP では，画像エンコーダーとテキストエンコーダーを一緒に訓練し，バッチ内の N 個の実ペアの画像とテキスト embedding のコサイン類似度を最大化する一方で，不正確なペアの embedding のコサイン類似度を最小化することで，マルチモーダルな embedding 空間を学習している．

V&A の分野おける最新の共有タスクに，V-WSD と呼ばれるものがある．これは，ある曖昧な語義を持つ単語と，その単語の語義の推定を補助する語句が与えられたとき，その単語の意図する語義に対応する画像を候補の中から選択するタスクである．与えられた目的の単語と候補の画像群の類似性を比較するには，言語と画像の両方の埋め込み表現を同一のベクトル空間上に得る必要がある．本研究では，そのような複数のモダリティを利用することができる事前学習モデルである，CLIP モデルを用いて，V-WSD タスクを行うことで，V&A 分野における CLIP の有用性を示した．また，与えられた曖昧な語義を持つ単語に対して，WordNet を活用して，同じく与えられた補助語句から語義の推定を行い，CLIP モデルの入力文として利用する手法を試みた．

実験では，V-WSD の trial データセットを用いて，全 16 問の問題に対して，WordNet を利用した目的単語の語義推定アルゴリズムを実装し，得られた語義定義文から，CLIP モデルへの複数のプロンプトを設定し，その正解率の差を評価した．考察では，実験結果と WordNet の語義推定の精度から，V-WSD タスクの問題に対して，それぞれ分析を行い，精度向上のための検討と課題の提示を行った．

Master's Thesis in Scholastic 2022, Major in Computer and Information Sciences,
Granduate School of Science and Engineering, Ibaraki University

Visual WSD with CLIP

Author : Youki Itou (21NM708A)

Adviser : Prof. Hiroyuki Shinnou

Abstract

In recent years, there has been a large expansion of research in the area of Vision-and-Language, both for images or videos and for natural language. CLIP is one of the pre-training models that can obtain embedded representations of the two domains of image and language. CLIP uses the Vision-Transformer for image encoding. CLIP trains the image and text encoders together to maximize the Cosine similarity of the image and text embeddings of the N real pairs in a batch, while minimizing the Cosine similarity of the embeddings of the incorrect pairs, thereby learning a multimodal embedding space by minimizing the Cosine similarity of the embedding of incorrect pairs.

One of the most recent shared tasks in the V&A area is called V-WSD. This is a task that, given a word with a certain ambiguous sense and a word that aids in the estimation of the word's sense, selects an image from a set of candidates that corresponds to the intended sense of the word. To compare the similarity between a given target word and a set of candidate images, it is necessary to obtain embedded representations of both the language and the images on the same vector space. We demonstrate the usefulness of CLIP in the V&A domain by performing the V-WSD task using the CLIP model, a pre-trained model that can exploit such multiple modalities. In addition, we tried a method of using WordNet to infer the meaning of words with ambiguous meanings from given auxiliary phrases, and using them as input sentences for the CLIP model. In the experiment, we implemented a WordNet-based word sense estimation algorithm for the target word for a total of 16 questions on the V-WSD trial dataset, set up multiple prompts to the CLIP model from the resulting word sense definition sentences, and evaluated the difference in their correct response rates. In the discussion, based on the experimental results and the accuracy of WordNet's word sense estimation, we analyzed the results for the V-WSD task problem, respectively, and discussed how to improve the accuracy and presented the issues.

目次

第 1 章

序論

近年，画像もしくは動画を対象とした研究と自然言語を対象とした研究は，それぞれ Computer Vision および Natural Language Processing の領域において，相互に影響しあいながら発展を遂げてきている．特に，深層学習の一種である Convolutional Neural Network（CNN）や Recurrent Neural Network（RNN）といった共通の機械学習手法が台頭し，それぞれの領域への参入障壁が低下している．その結果，視覚と自然言語を融合する研究が様々な広がりを見せている．

Vision-and-Language における主要な研究分野は，例えば，

- 入力した画像の内容を自然な文として記述するタスクである Image Captioning
- ある画像とその画像に関する質問を提示されたときに，正しい答えを導き出すタスクである Visual Question Answering
- 入力されたテキスト（キャプション）を条件として，そのテキストに合う画像を生成するタスクである Text-to-Image Generation
- クエリ文あるいはクエリ文 + 元画像を基にデータベースから条件に合う画像を検索するタスクである Image Retrieval

といったものが存在している．

Vision-and-Language の研究分野の一つに，Visual Word Sense Disambiguation（V-WSD）がある．これは，ある単語と限られた文脈が与えられたとき，その単語の意図する意味に対応する画像を候補の集合から選択するタスクであり，2022 年 7 月の SemEval

[*1]-2023 において発表されたマルチモーダルな共有タスクである．V-WSD の問題の具体例を以下の図 1.1 に示す．

図 1.1: 問題例．曖昧な目的単語 andromeda を含む完全なフレーズ「andromeda tree」と，以下の 10 枚の候補画像が与えられたとき，対応する画像（Gold image）を選択するタスクである．この場合，正しい画像は以下のように左の 1 枚目である．

V-WSD では，与えられた目的単語と候補画像群の類似性を比較するには，言語と画像の両方の埋め込み表現を得る必要がある．そのような複数のモダリティを利用すること

[*1] 語義評価会の Senseval から発展した，計算意味解析システムの評価会．自然言語処理（NLP）の国際的な研究ワークショップを開催し，意味解析の現状を改善し自然言語の意味論においてますます困難になっている様々な問題において，高品質のアノテーションデータセットの作成を支援することを使命としている．

ができる事前学習手法の一つに Contrastive Language-Image Pre-training（CLIP）[1] がある．CLIP は，2021 年 1 月に OpenAI [*2]によって公開され，事前学習済みモデルが提供されている．CLIP は Web 上に豊富にある画像とテキストのペアのみの学習をしており，ImageNet [*3]やその関連データセットで，高い精度での分類を zero-shot で予測することが可能なモデルである．この CLIP を用いることで，言語と画像の類似性を簡単に比較することが可能になる．

しかし，この V-WSD では，目的単語には通常，複数の語義が存在しており，候補となる画像群にも問題の正解に等しい目的単語の語義とは，異なる語義に一致した画像も含まれている．したがって，与えられた目的単語の語義曖昧性を解消し，問題の正解に沿った語義を定義する必要が出てくる．

本研究では，V-WSD において与えられた目的単語の語義を正しく定義し，CLIP による言語と画像の類似性の推論精度を向上させるための手法を提案する．与えられた目的単語とその補助語句による目的単語の語義定義文は，各々の画像との組合せによって，CLIP が推論した類似度に影響を与えることがある．よって，各問題に対して語義の定義法と類似度の推論方法を変更することで，全体の推論精度を向上させることができると考える．

[*2] 人工知能を研究する非営利団体．人類全体に，害をもたらすよりは，有益性があるやりかたで，オープンソースと親和性の高い人工知能を，注意深く推進することを目的として掲げている．

[*3] 物体認識ソフトウェアの研究で用いるために設計された大規模な画像データベース．WordNet の階層構造（現在は名詞のみ）に沿って整理された画像データベースで，階層の各ノードが数百，数千の画像で描かれている．

第 2 章

関連研究

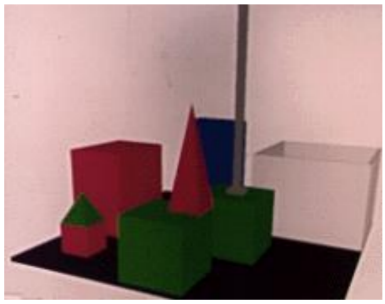
Vision-and-Language という分野名は，深層学習の普及に伴って呼ばれるようになったが，それ以前より画像と言語の両方を扱う研究は古い歴史があり，自然言語とキーボード，マウス入力を組合わせてイラストを描画させる研究 [2] や手描きの絵から説明文を生成する研究 [3]，自然言語による物体操作と画像付き質問応答の研究 [4] が存在している．（図 2.1）

従来の Vision-and-Language の研究分野では，実世界上の多様な物体やテキストを記号的に定義し，計算の土台の載せることが困難であり，したがって，タスクをなるべく限定する必要があった．しかし，現在はインターネットの普及に伴い扱える画像やテキストのデータ量が増えた事，高性能な計算機が開発された事，統計的手法の発展した事によって，画像や言語もベクトル化することが可能になった．Transformer をはじめとした大規模パラメータやデータセットを用いる学習手法とそれを実現する高性能な計算機が登場など，深層学習の発展・普及に伴い，Vision-and-Language のための様々な学習済みモデルが生まれ，これらを利用した応用研究も増加している．

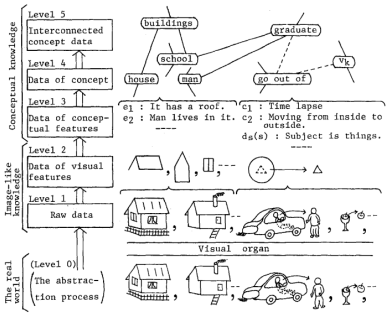
2.1 画像における zero-shot 分類の手法

2.1.1 zero-shot 学習

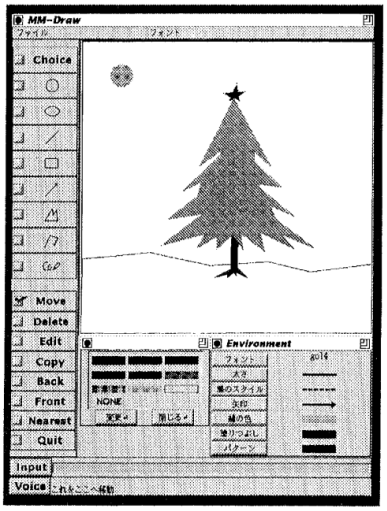
コンピュータビジョンの分野では，ディープラーニングの手法が応用計算と機械知能の両方で大きな成果を上げている．特に，ディープラーニングは画像分類において多くの成功を収めている．十分な量のラベル付けされた学習データが存在する場合，ディー



(a) 自然言語による物体操作と画像付き質問応答



(b) 手描きの絵から説明文を生成



(c) 自然言語とキーボード，マウス入力を組合わせてイラストを描画

図 2.1: Vision-and-Language の古典的研究

プニューラルネットワークを利用することで，様々なタスクにおいて機械は人間に近いレベル，あるいは人間を超えるレベルでパフォーマンスを発揮することができる．しかし，従来のディープニューラルネットワークモデルは，優れた性能を実現するために重要な要素に依存している．一般に，ディープニューラルネットワークは学習のために膨大な数のラベル付けされた学習データを必要とするが，大量のサンプル収集とラベル付けは困難で時間がかかってしまう．

画像分類タスクにおいて，以下に示すいくつかの場合では，学習済みモデルを微調整することがしばしば困難になる．

- カテゴリクラスの数が膨大である場合．例えば，人間は約 3,000 の基本レベルのクラスを区別することができ，各基本クラスは，異なる犬種のように下位のクラスとして拡張することができる．このように膨大な数のカテゴリが存在する状況では，各カテゴリが十分な数のラベル付きサンプルを持つタスクを構築することは困難である．
- サンプル数が少ない希少なクラスが存在する場合．花や鳥の細かい分類や，ある特定の状況に対応する医療画像など，対応するサンプルを得ることが困難な希少なクラスが存在する場合．
- 時間の経過と共に候補となるクラスが増加する場合．例えば，新しく収集したメディアデータから新しいイベントを検出するタスク，製品のブランドを認識するタスク，いくつかの文体を学習するタスクなど，データセットが急速に変化し，時間の経過とともに候補クラスが増加する．

このようないくつかの状況下では，ターゲットとなるデータセットで学習済みモデルを再学習することは，あまり適切ではないことが多く，学習済みモデルを微調整するのは，ラベル付けされたターゲットのサンプルのいくつかが得られる場合にのみ可能である．

このような制約を克服するために，zero-shot 学習は，人間の学習能力をシミュレートするために設定されている．馬の形，縞の概念，黒と白の色などの知識を備えた子供が，「シマウマは白と黒の縞で覆われた馬に似ている」と言われれば，初めて見るシマウマでも認識できる可能性は高い．

一般的に，zero-shot 学習は，各カテゴリを説明するための補助情報と，それに対応するいくつかのサンプルを基に，サンプルと補助情報の相関を構築するモデルを学習する

ことにより，補助情報と相関を基に，未知のカテゴリに対する分類を拡張することが可能となる．

2.1.2　学習のシナリオ

従来の画像分類タスクでは，訓練集合とテスト集合のインスタンス分布の違いにより，訓練したモデルがテスト時に訓練集合の時と同じような性能を発揮することができない．この現象は zero-shot 学習でも見られ，さらに，既知クラスと未知クラスが不連続であるため，より深刻である．このような，既知クラスと未知クラスの分布の違いをドメインシフトと呼ぶ [5]．さらに，このようなモデル性能の低下はクラスレベルのオーバーフィットと呼ばれる [6]．これらの課題に対し，サンプルや補助情報から得られる分類知識を効果的に用いることで，学習やテストを含む様々な段階で知識を導入する方法が提案されている．その結果，実装の場面は多様化している．

zero-shot 学習では，サンプル空間と補助情報空間の両方が定義可能であり，それに応じてシナリオを分けることができる．

一般に，学習段階から見ると，inductive，semantic transductive，transductive の 3 つのシナリオに分けることができ，それぞれ以下のように定義される．

- **Inductive zero-shot 学習**．ラベル付けされた学習サンプルと，既知クラスの補助情報のみが学習中に利用可能な zero-shot 学習．
- **Semantic transductive zero-shot 学習**．ラベル付けされた学習サンプルと全クラスの補助情報が利用可能な zero-shot 学習．
- **Transductive zero-shot 学習**．ラベル付き学習サンプル，ラベルなしテストサンプル，および全クラスの補助情報が学習時に利用可能な zero-shot 学習．

この定義から，Inductive zero-shot 学習は，対象クラスとインスタンスの両方が未知であるため，最も難しい学習シナリオであることがわかる．このシナリオで学習したモデルは，クラスレベルのオーバーフィッティングに陥る可能性が高い．これに対し，残りの 2 つのシナリオで学習したモデルは，分類知識が未知の情報によって導かれるため，明確な学習目的を共有する．

しかし，これらの学習したモデルは，Inductive シナリオで学習したモデルほどには，新しい未知のクラスに対して汎化しない．zero-shot 問題が提案された初期段階では，従

来の zero-shot 学習として知られる，未知クラスで良好な分類を達成することのみに焦点を合わせていた．その後，既知のカテゴリも分類の候補に含めると，未知クラスの分類が壊滅的な打撃を受けることが判明した．つまり，初期の提案モデルでは，既知のカテゴリと未知のカテゴリをうまく区別できず，新しいクラスの認知概念を構築することができなかった．その結果，既知クラスと未知クラスの両方を分類する必要がある，一般化 zero-shot 学習と呼ばれるより困難な課題が注目されるようになった．

zero-shot 学習の本来の目的は，サンプルがない状態で，学習した知識と支援情報からクラスの認知概念を構築する人間の過程をシミュレートすることである．しかし，構築された認知概念は，既知クラスと未知クラスが正しく区別されて初めて正確に評価されるため，現在の研究の焦点は一般化されたものに移っている．

2.1.3　問題の定義

ここで，zero-shot 学習の定義を示す．zero-shot 学習では，各サンプルは各ピクセルに値を保持するテンソル形式で特定のオブジェクトを含む画像として設計されている．より簡便に実装するために，画像を用いる代わりに，事前に学習したディープニューラルネットワークによって抽出された視覚的特徴をサンプルと見なすことが一般的である．

ここでは，厳密な表現として，画像全体を入力サンプルとする．K 個のクラスから全部で N 個のサンプルがあるとして， 既知のクラスと未知のクラスの両方からの全ての画像サンプルの集合を $X = X^S \cup X^U$ とし，画像 x_i の特徴 $F(x_i)$ を得るための特徴抽出器を $F(\cdot)$ と表現する．

同様に，対応するラベル集合は $Y = Y^S \cup Y^U$ と表し， $y_i = k$ はサンプル x_i が k 番目のクラスに属することを示す．

補助情報の集合は $A = A^S \cup A^U$ と表され，K 個のベクトルを含み，各ベクトル a_k は k 番目のクラスの補助情報を表す．

ここで，K^S と K^U はそれぞれ既知クラスと未知クラスの数を示し，A で表される最初の K^S クラスは便宜上，既知クラスと仮定する．既知クラスと未知クラスは不連続であり，$X^S \cap X^U = Y^S \cap Y^U = A^S \cap A^U = \emptyset$ となる．

既知クラスのサンプルの一部は，学習過程に参加しないテストインスタンスとして用いられるため，既知クラスのサンプルとラベルの集合は，さらに $X^S = {X^S}_{tr} \cup {X^S}_{te}$ と $Y^S = {Y^S}_{tr} \cup {Y^S}_{te}$ として学習とテストの集合に一貫して分けられる．具体的には，学習

用とテスト用の既知集合は，共に K^S の全ての既知クラスをカバーする必要がある．

学習プロセスには 3 つのシナリオがあるため，学習セット $D_{tr} = \{X_{tr}, Y_{tr}, A_{tr}\}$ はそれぞれ inductive，semantic transductive，transductive シナリオに対して ${D^I}_{tr} = \{{X^S}_{tr}, {Y^S}_{tr}, A^S\}$，${D^S T}_{tr} = \{{X^S}_{tr}, {Y^S}_{tr}, A\}$, ${D^T}_{tr} = \{{X^S}_{tr} \cup X^U, {Y^S}_{tr}, A\}$ という 3 つの形式で定義できる．

また，テストセット $D_{te} = \{X_{te}, Y_{te}, A_{te}\}$ に対して，それぞれ従来型 zero-shot 学習では ${D^C}_{te} = \{X^U, Y^U, A^U\}$，一般化 zero-shot 学習では ${D^G}_{te} = \{X^U \cup {X^S}_{te}, Y^U \cup {Y^S}_{te}, A\}$ の 2 つの形式で定義することができる．

これらの定義により，zero-shot 学習の目標は，情報抽出器 M（特徴抽出器 $F(\cdot)$ を含む）を学習集合 D_{tr} 上で定まった，あるいは学習可能な分類器 C を用いて学習させ，X_{te} に対する分類を実現することと表すことができる．

2.1.4　zero-shot 画像分類の手法

Yang らの調査 [7] より，zero-shot 画像分類の各手法は，情報抽出器の設計に基づき，埋め込み型手法（Embedding methods）と生成型手法（Generative methods）の 2 つに大別でき，図 2.2 に示すような分類構造が提案されている．

埋め込み型手法（Embedding methods）　embedding に基づく手法の主な目的は，ディープニューラルネットワークを用いて 学習される投影関数を用いて，画像の特徴や意味的属性を共通の埋め込み表現に マッピングすることである．共通の埋め込み空間は，視覚的空間，意味的空間，あるいは 新たに学習された中間的空間とすることができる．

生成型手法（Generative methods）　生成法の中核を成すのは意味情報を入力とし，対応する擬似サンプルを出力する生成器である．このような生成器は，変分オートエンコーダ（VAE）[8] や敵対的生成ネットワーク（GAN）[9] のアーキテクチャに基づいて構築することができる．また，対応するセマンティクスを持つラベル付けされたサンプルで学習させることもできる．そして，未知のセマンティクスを用いることで，未知のクラスの擬似サンプルを生成し，zero-shot 学習課題を一般的な分類に変換することができる．

本研究で扱う CLIP モデルは，画像の zero-shot 分類では埋め込み型手法に該当する．

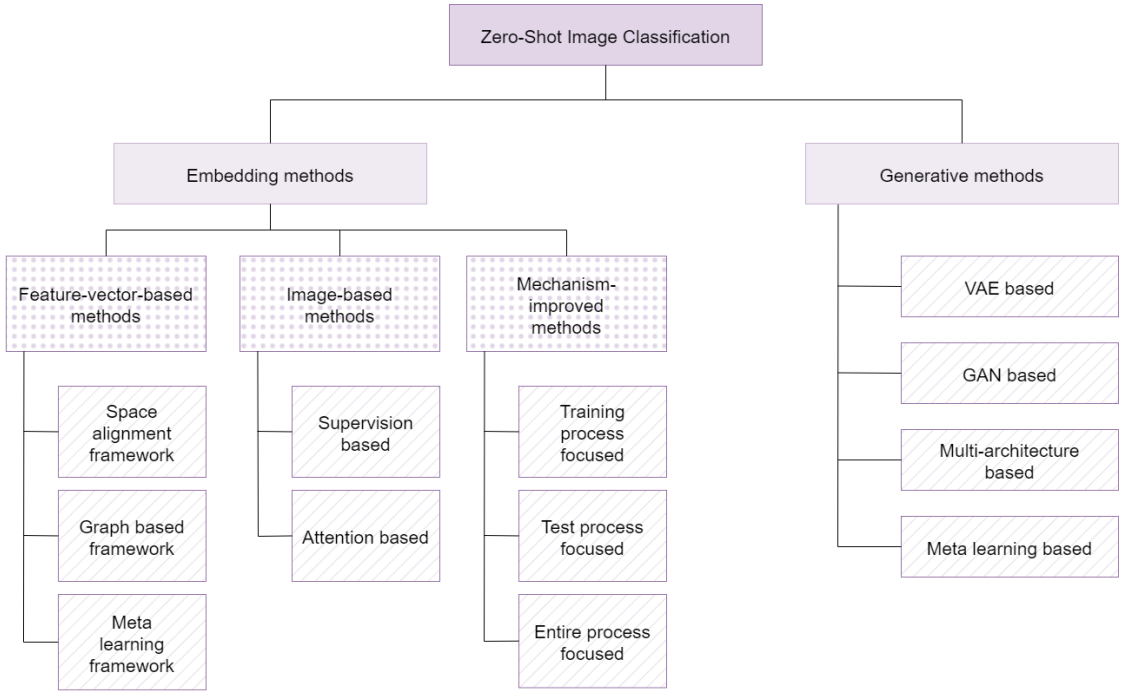


図 2.2: zero-shot 画像分類法の分類体系.

（出典：論文 [7] Fig 2. を参考に作成）

2.2　画像からのキャプション生成

視覚と言語を結びつけることは，Generative Intelligence（生成的知性）において重要な役割を果たす．このため，画像キャプション，すなわち構文的に意味のある文章で画像を説明するための様々な研究が行われてきた．2015 年以降，このタスクは一般的にビジュアルエンコーダーとテキスト生成のための言語モデルで構成されるパイプラインで対処されてきた．この間，オブジェクト領域，属性の利用，マルチモーダル接続の導入，Attention や BERT のような事前学習手法を通じて，2 つの領域を横断する研究は大幅に進化している．

画像のキャプション生成は，視覚理解システムと構文的に正しい文を生成できる言語モデルを用いて，画像の視覚的内容を自然言語で記述するタスクである．

脳科学の研究により，人間の視覚と言語生成の関連性が明らかになったのはここ数年である．同様に，人工知能においても，画像処理と言語生成が可能なアーキテクチャの設計はごく最近の事柄である．これらの研究の目的は，入力画像を処理し，その内容を表現

し，言語の流暢性を維持しながら視覚的要素とテキスト要素の間の接続を生成してそれを一連の単語に変換するための最も効果的なパイプラインを見つけることである．

画像キャプション生成の初期に提案されたアプローチは，記述検索またはテンプレート充填と人手で作成する自然言語生成を含んでいた．画像キャプション生成は現在，深層学習ベースの生成モデルの利用を基本としている．その標準的な構成では，タスクは入力がピクセルである画像からシーケンスへの問題である．これらの入力は，視覚的エンコーディングのステップで 1 つまたは複数の特徴ベクトルとしてエンコードされ，言語モデルのための入力が準備される．そして，与えられた語彙に従ってデコードされた単語やサブワードの列を生成している．

この数年間で，画像キャプション生成のためのモデル設計は大幅に改善されている．グローバルな画像記述子を供給する Recurrent Neural Network（RNN）を採用した最初の深層学習ベースの提案から，Transformer と Attention 構造，BERT に似たアプローチや強化学習を活用した手法へと変化してきている．

Stefanini らの調査 [10] より，現在の画像キャプション生成における，画像エンコーディングのアプローチは，以下の 4 つの手法に分類できる．

- グローバル CNN 特徴に基づく，**non-attentive methods**.
- グリッドまたは領域のいずれかを用いて視覚コンテンツを埋め込む，**additive attentive methods**.
- 視覚領域間の視覚的関係を加える，**graph-based methods**.
- Transformer ベースの機構を用いる，**self-attentive methods**.

本研究で扱う CLIP モデルは，画像のキャプション生成における画像エンコーディングの手法は，4 番目の self-attentive methods に該当する．

第 3 章

CLIP: Contrastive Language-Image Pre-training

3.1 導入と概要

Vision-and-Language における学習済みモデルの一つに CLIP と呼ばれるモデルがある．CLIP は Web 上に豊富に存在する画像とテキストのペアのみから学習を行っており，ImageNet 等のベンチマークデータセットで，高い精度での分類を zero-shot で予測することが可能なモデルである．

自然言語分野では，事前学習を未加工の文章から直接学習することが試みられており，GPT-3 に代表されるような優れた zero-shot 転移を可能にしている．CLIP は，画像処理分野でも未加工の画像とテキストから直接学習することはできないのか，というモチベーションから行われている．同様の発想から VirTex [11] や，ICMLM [12], ConVIRT [13] などがテキストから画像表現を学習するモデルとして発表されている．

しかし，未加工の自然言語の教師データをそのまま画像表現の学習に利用する試み自体は盛んであるとはいえない．何故なら，総じてベンチマークテストに対して既存の他の方法より精度が低いからである．この未加工の自然言語を付随しただけの教師なし学習と普通のラベル型教師あり学習との間を補完する形で弱教師学習も行われている．例えば，Instagram 画像上の ImageNet 関連のハッシュタグを予測する学習などがあげられる．これらの試みは，自然言語の汎用性を活かしてより広範囲の視覚的概念を表現する教師ラベルとして自然言語を利用している．弱教師モデルは比較的よい精度をベンチマークテストで出すことに成功しているが，予測を実行するために静的なソフトマック

ス分類器を使用しており，動的な出力メカニズムを欠いている．これは，柔軟性を著しく制限し zero-shot の能力を制限することにつながっている．

CLIP の開発者達は，弱教師モデルとこれまでの自然言語から直接画像表現を学習するモデルに性能の差を生み出している決定的な違いは，データセットの規模にあると考えた．前者が 100 万〜10 億という単位で画像を利用しているのに対して，後者はおよそ 20 万程度であった．CLIP では，この差を埋めて学習した際の自然言語型教師モデルの可能性を探ることが目的の一つだった．結果，利用する学習データとしてインターネットから画像とテキストを集め，4 億組の大規模なデータセットを構築している．そのため，モデルの改良は主眼にはなく，利用しているモデルは，VerTex を参考に，ConVIRT をよりシンプルにしたモデルとなっている．

3.2　モデル構造

CLIP のモデル構造は，VirTex と同様の CNN とテキスト Transformer ではなく，より効率性の良い VisionTransformer [14] を利用している．正確な言葉を予測しようとすると非常に難しく，一方で対象表現学習でも同程度の精度がだせることが最近の研究で分かってきており，また，生成モデルは学習により多くのデータが必要になることも分かっている．以上の点から，テキスト中の正確な単語ではなく，テキスト全体がどの画像とペアになっているかだけを予測するという，より簡単な代理タスクを解決するためのモデルとなった．Bag-of-Words エンコーディングのベースラインに対して，予測目的単語を対照目的単語に置き換えたところ，ImageNet への zero-shot 転送率がさらに 4 倍に向上したことが確認されている．

最終的なモデルは，N 個のペアのバッチが与えられると，CLIP は，バッチ全体で $N \times N$ 個の可能性のあるペアリングのうち，どのペアリングが実際に発生したかを予測するように訓練される．実際に CLIP では，画像エンコーダーとテキストエンコーダーを一緒に訓練し，バッチ内の N 個の実ペアの画像とテキストの embedding のコサイン類似度を最大化する一方で，不正確なペアの embedding のコサイン類似度を最小化することで，マルチモーダルな embedding 空間を学習している．（図 3.1 参照）

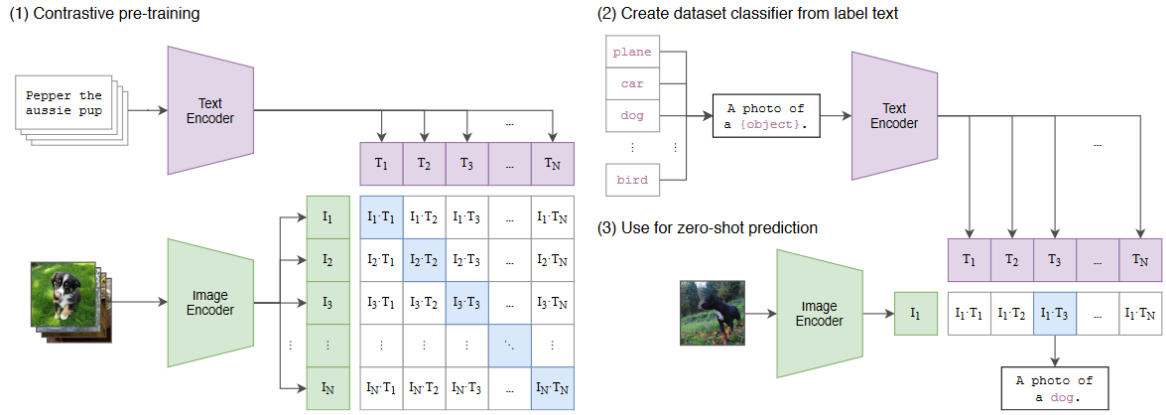


図 3.1: CLIP の事前学習時（左）と推論時（右）の全体像.

（出典：論文 [1] Figure 1. より）

3.2.1　モデルの詳細

画像エンコーダ

モデル 1（比較ベースモデル） ResNet-50 [15] の改良版を利用．グローバル平均プーリング層をアテンションプーリング機構に置き換え．アテンションプーリングは，クエリが画像のグローバル平均プーリングされた表現に基づいて条件付けされる「Transformer スタイル」のマルチヘッド QKV アテンションの単一レイヤとして実装されている．

モデル 2（より洗練されたモデル） 基本的に VisionTransformer と同一のものを利用．違いは，Transformer の前にパッチと位置の埋め込みを組み合わせたものに追加のレイヤ正規化を追加し，僅かに異なる初期化スキームを使用した点のみ．

テキストエンコーダ

基本的に Transformer を利用．ベースサイズとして，8 つのアテンションヘッドを持つ 63M パラメータの 12 層 512 ワイドモデルを使用．49,152 個の vocab サイズを持つテキストの小文字のバイトペアエンコーディング（BPE）表現で動作する．計算効率のため，最大シーケンス長は 76 に制限．テキストシーケンスは [SOS] と [EOS] トークンで括られ，[EOS] トークンにおけるトランスフォーマーの最上位層の活性化は，層を正規化した後，マルチモーダル埋め込み空間に線形投影されたテキストの特徴表現として扱われる．事前に学習した言語モデルで初期化したり，補助的な目的として言語モデルを追加したりする能力を維持するため

に，Masked self-attention が使用されている．

3.2.2　事前学習

事前学習させたモデルは，以下の 8 つになる．

- ResNets × 5（ResNet-50，ResNet-101，RN50x4，RN50x16，RN50x64）
- Vision Transformers × 3（ViT-B/32，ViT-B/16，ViT-L/14）

なお，ViT-L/14 については，性能を向上させるために，より高い 336 ピクセルの解像度で 1 つのエポックを追加して事前学習が行われている．このモデルを Radford らの論文 [1] では ViT-L/14@336px と表記しています．ViT-L/14@336px が最も性能が高く，本研究で「CLIP」とされているものは基本的にこのモデルを指している．

エポック数

すべての学習は 32 エポック行われた．最大の ResNet モデルである RN50x64 は，592 個の V100 GPU 上で 18 日間，最大の Vision Transformer は 256 個の V100 GPU 上で 12 日間かかっている．

最適化関数

Adam（重み減衰正規化とコサイン関数を利用した学習率の減衰を組み合わせたもの）を利用している．

ハイパーパラメータ

計算量を制約するためにヒューリスティックに適応させている．

学習可能な温度パラメータ τ

0.07 に相当するものに初期化され，学習の不安定性を防ぐために必要であることがわかった 100 以上でロジットをスケーリングしないようにクリップされた．

ミニバッチ

32,768 という非常に大きなミニバッチサイズを使用している．

その他

トレーニングを高速化し，メモリを節約するために，Mixed-precision という手法と，追加のメモリを節約するために，gradient checkpointing，half-precision Adam statistics，半確率的に丸められたテキストエンコーダの重みが使われて

いる．

3.3　CLIP の zero-shot 転移の性能

ここでは，Radford らの論文 [1] で行われた，CLIP モデルの zero-shot 転移の性能を評価した実験の詳細を記述する．

一般的に，zero-shot 転移とは，未知のオブジェクトカテゴリーに対する画像分類のことを指すが，ここでいう zero-shot 転移とは，未知のデータセットに対する画像分類の意味として用いる．

基本的に zero-shot 転移や教師なし学習は，表現学習として注目されてきたが，CLIP は zero-shot 転移学習をタスク学習能力を測るものとしてみなしている．ただし，ベンチマークテストで使われるデータセットは実践のタスクに向いているとは言い難く，タスク一般化能力ではなく分布移動やや領域一般化に対するロバスト性を測るためのものとしている．

CLIP モデルと Visual N-Grams モデルの zero-shot 転移の性能を比較したところ，大幅に精度が改善しているという結果になっている．特に，ImageNet データセットでは，Visual N-Grams モデルが 11.5% なのに対し，CLIP モデルは 76.2% と精度が改善している．

CLIP モデルの zero-shot の精度はほぼ ImageNet で学習した ResNet-50 と同水準になっている．また，トップ 5 の結果は 95% にも及ぶ．

3.3.1　zero-shot 転移に対する CLIP の利用法

CLIP は，画像と文章のペアを予測するモデルである．各データセットについて，データセット内のすべてのクラスの名前を潜在的なテキストペアリングの集合として使用し，CLIP に従って最も確率の高い（画像，テキスト）ペアを予測する．

始めに，画像の特徴埋め込みと，それぞれのエンコーダによる可能性のあるテキストのセットの特徴埋め込みを計算する．次に，これらの埋め込みのコサイン類似度を計算し，温度パラメータ τ でスケーリングし，ソフトマックスを用いて確率分布に正規化する．この予測層は，L2 正規化された入力，L2 正規化された重み，バイアスなし，温度スケーリングを持つ多項ロジスティック回帰分類器である．

CLIP の事前学習の各ステップは，クラスごとに 1 つの例を含み，自然言語記述によって定義された総クラス数 32,768 のコンピュータビジョンデータセットに対して，ランダムに作成されたプロキシの性能を最適化していると見ることができる．zero-shot 評価のために，テキストエンコーダによって計算された zero-shot 分類器を一度キャッシュし，その後のすべての予測のために再利用します．これにより，zero-shot 分類器の生成コストをデータセット内の全ての予測値に対して償却することが可能となる．

3.3.2　プロンプトエンジニアリング

zero-shot でベンチマークデータセットを利用する上で，以下の問題が発生している．

多義語だけ問題

文章がない単語だけのカテゴリーリストでは，多義語の意味が定まらないという問題．普通のデータセットでも，別のクラスに分類されていながら同一の単語のものが存在している．(例: cranes（動物の鶴と重機のクレーン），boxer（動物の犬種とスポーツ選手))

単語だけ問題

CLIP の事前学習時に単語だけというケースが珍しく，与えるカテゴリーリストが単語だけでは高い精度での予測ができないという問題．

ここでは，与えるプロンプトを工夫することで上記の問題に対応している．始めに，プロンプトを'a photo of a label' と単語ではなく，長くすることで上手く機能することが判明した．ImageNet の場合，1.3% の性能の改善に繋がっている．また，'a photo of a label, a type of pet' など，ベンチマークデータセットのタイプがわかっている場合は情報を追加で表現することで多義語の問題に対応している．この応用はほかにも適用することができ，OCR データセットでは，認識したいテキストや数字の周りに引用符を付けることや，衛星画像の照合データセットでは画像がどの形式のものであるかを特定できる'a satellite photo of a label' といった形にするなどの工夫で精度が上昇することが確認されている．

加えて，プロンプトのアンサンブル学習（80 の異なるプロンプトを利用）することで 3.5 %の精度上昇が確認された．結果，両方の利用でおおよそ 5 %の上昇を実現できることが分かっている．

3.3.3　パフォーマンス分析

Zero-shot CLIP モデルと各データセットで学習された ResNet-50 のパフォーマンスを比較分析の結果を，以下の図 3.2 に示す.

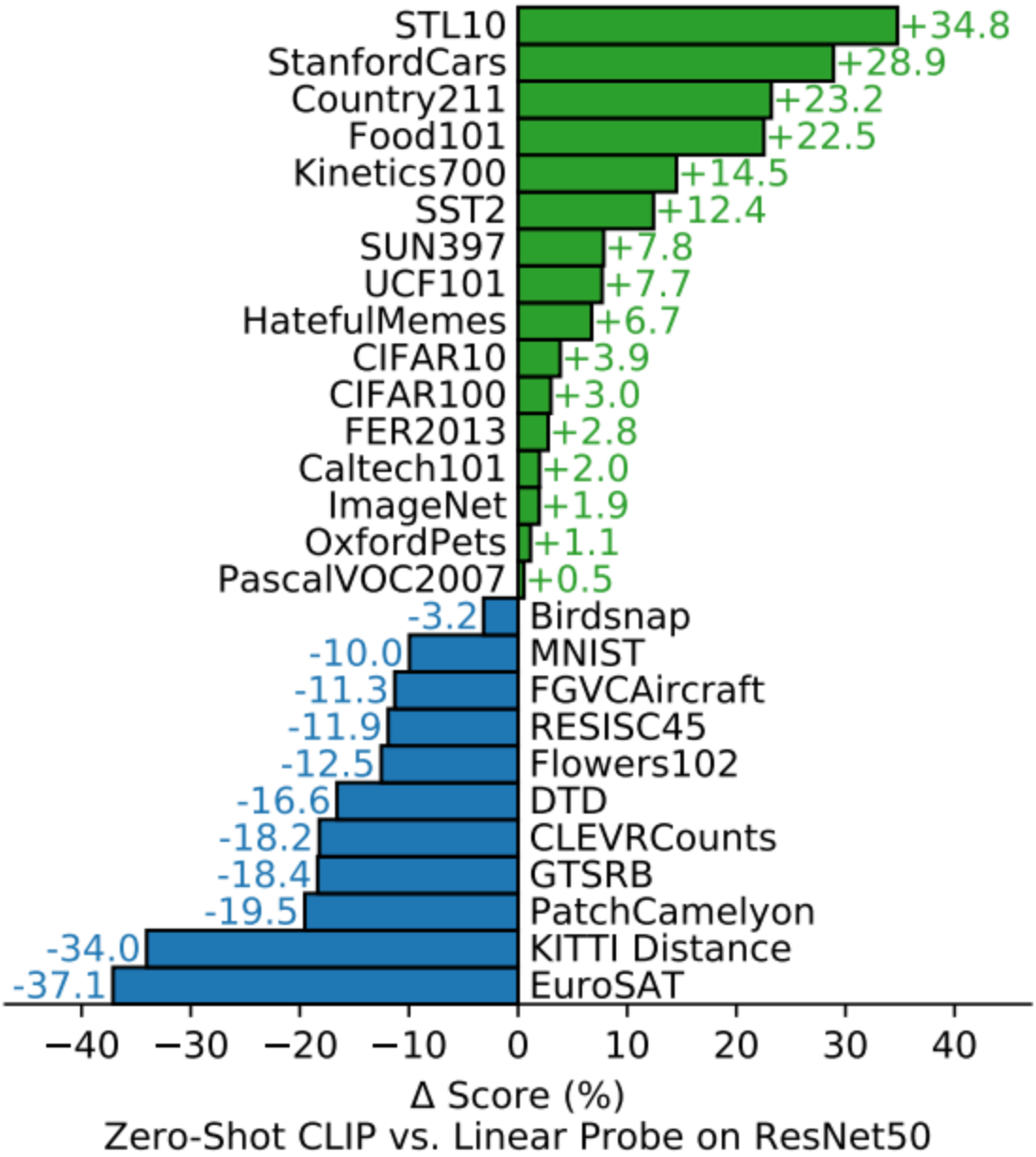


図 3.2: Zero-shot CLIP と ベースラインモデルの比較.

（出典：論文 [1] Figure 5. より）

ResNet-50 は ImageNet で学習されており，各データセットの画像の特徴量を求め，その特徴量に対してロジスティック回帰モデルにより教師ラベルを使って fine-tuning を

行う．特徴量は，ResNet-50 により計算される特徴量を使い，ロジスティック回帰のパラメータのみを学習している．

図 3.2 には，CLIP の精度から ResNet-50 の Linear probe の精度を引いた数値が載っている．緑のグラフは，CLIP の精度の方が良いデータセットで，青のグラフは，ResNet-50 の精度の方が良いデータセットとなっている．

全体としては，25 データセット中 16 データセットで CLIP の方が精度が良くなっているが，一貫してどちらかの方が良いというわけではなく，データセットにより結果は大きく異なっている．その理由として，結果が大きく違う理由について，事前学習データとタスクデータの違いが挙げられている．

ImageNet，CIFAR-10，CIFAR-100 といった，一般的なデータセットについては，どちらのモデルも事前学習で見たことのある画像データセットであったため，有意な差は無かった．

Kinetics700 や UCF101 データセットでは CLIP が大きく上回っているが，これらのデータセットは動画から切り取った画像データセットであり，ラベルが動詞などになっている．よって，自然言語によるラベルデータを使った CLIP の方が，名詞ベースのラベルデータを使った ImageNet よりも精度が良くなっていると考えられる．

一方で，CLIP の精度が大きく下回っているデータセットは抽象度が高く，複雑な画像サンプルになっている．EuroSAT や RESISC45 は衛星画像で，PatchCamelyon はリンパにある腫瘍を見つけるデータセットである．これらを zero-shot の設定で解くというのは，人間にとっても専門知識が無ければ困難なタスクであると考えられる．

他にも，CLEVRCounts は物体の数を数えるタスク，GTSRB は自動運転の画像，KITTI Distance は一番近い自動車間の距離を予測するタスクになっており，これらのデータセットでは，教師あり学習の方が良くなっている．

これらのタスクは人間にとっては，それほど難しいタスクではないとされているため，CLIP モデルの今後の課題とされている．

第 4 章

CLIP と WordNet による V-WSD

CLIP はインターネットを利用して構築された，画像と画像を説明する自由テキストのペアのデータセットである，巨大な 4 億組もの自然言語教師データ「WebImage Text」を利用している．また，カテゴリーを利用者側で自由に設定できる自然言語教師型画像分類モデルであり，多様なタスクに対して zero-shot 転移で優れた精度を発揮する．そこで，Vision-and-Language の研究分野における最新の共有タスクである V-WSD に対して，自然言語教師型画像分類モデルである CLIP を活用することで，自然言語処理における意味解析の現状の改善を試みる．

V-WSD は，ある単語と限られた文脈が与えられたとき，その単語の意図する語義に対応する画像を候補の集合から選択する課題である．ここで与えられる単語は，複数の曖昧な語義を持つことが定められており，また，与えられる候補画像の集合内にも，目的単語のそれぞれの語義に対応したものが含まれている．例えば，図 1.1 の問題例のように，目的単語「andromeda」は，一般的に，

1. ギリシャ神話に登場する女性の名前.
2. 1. から派生した，夜空に浮かぶ星座あるいは銀河の名前.
3. ツツジ科アセビ属に属する常緑性の低木．アセビ.

の異なる 3 つの語義を持っており，候補となる画像群にも，それぞれの語義に合致した画像が含まれていることが見て分かる．

V-WSD タスクでは，与えられる目的単語とその単語の語義の推定を補助するフレーズ（補助語句）から，問題における目的単語の正しい語義を推定しなければならない．CLIP モデルを用いることで，候補となるそれぞれの画像と目的単語あるいはフレーズ

から，類似度を計算することは容易に行える．しかし，仮に CLIP モデルがその単語の全ての語義を理解していたとしても，候補画像群から正答を選び出すことは困難である．何故なら，目的単語の正答ではない語義とその語義に対応した正答ではない画像のペアの類似度が，正答のペアよりも高くなることが起こるからである．

そこで，複数の曖昧な意味を持つ目的単語の語義を正しく推定するために，WordNet を用いて語義推定を行った．WordNet は英語の大規模な語彙データベースであり，名詞，動詞，形容詞，副詞が，それぞれ異なる概念を表す認知同義語の集合（synsets）にグループ化されている．

WordNet は，シソーラスと呼ばれる単語辞書を使って単語間の類似度を計測することができる．シソーラスは，似ている単語同士は近くに，そうでない単語同士は遠くに位置するような辞書である．これを用いることで，V-WSD において，目的単語とそれを補助するフレーズから，2 つの語句の間の語義同士の類似度を計測することができる．ここで，フレーズを構成する目的単語とその補助語句の語義の類似度が，他の語義の組合せよりも高くなると仮定すると，2 つの語句の間の語義同士の類似度が最も高い組合せを，それぞれの正答の語義であると導ける．

WordNet には，単語の各語義に対して定義文が定義されている．この目的単語に対して導出した語義の定義文を用いることで，CLIP モデルに対して，候補画像の説明文として適切なテキストを入力することが可能になる．

本研究の提案手法（図 4.1）をまとめると，以下のようになる．

1. フレーズを構成する 2 つの語句の語義間の類似度が，最も高くなると仮定する
2. 仮定より，2 つの語句の語義の最高類似度の組合せを，目的単語と補助語句の語義とする
3. 各問題の目的単語と補助語句から，2 つの語句の間の類似度を計算する
4. 推定した目的単語の語義定義文を CLIP モデルへの入力文として利用する
5. CLIP モデルを用いて，V-WSD タスクを解く

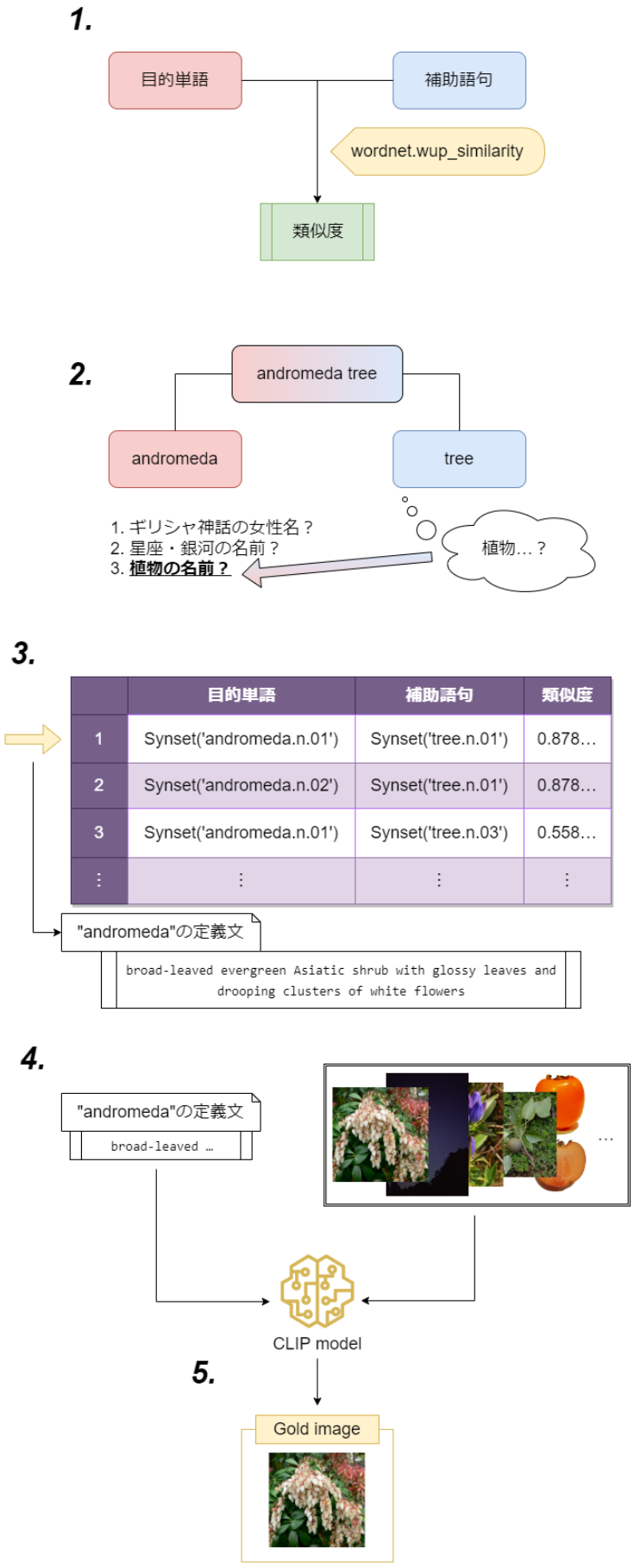


図 4.1: CLIP と WordNet による V-WSD

第 5 章

実験

5.1 実験設定

本実験では，WordNet を用いて目的単語の語義定義文を推定して，CLIP モデル の入力文にし，V-WSD タスクを解かせ，その正解率を測定した．提案手法において，WordNet を用いた目的単語と補助語句の各語義の組合せに対する類似度の計算法は，Wu-Palmer アルゴリズムを用いた．Wu-Palmer アルゴリズムは，2 つの単語に共通し，かつ最も近い上位概念の root からの深さ N と，その共通する概念からの各単語への深さ N1 と N2 を用いた計算法である．Wu-Palmer アルゴリズムで計算した値は 0 から 1 の範囲となり，0 が最も類似度が低く，1 が最も高い．

5.1.1 実験データ

V-WSD タスクで使用するデータセットは，以下の URL で公開されている．

`https://raganato.github.io/vwsd/`

このデータセットには，モデル訓練用の train データと試験用の trial データの 2 種類が存在している．今回は CLIP モデルの fine-tuning は行わないため，train データは使用せず，16 個の問題で構成された trial データを用いて，正解率を測定した．

データセットは，目的単語，全フレーズ，候補画像のファイル名群の 3 つが空白で区切られた形式で 1 行ごとに書かれているテキストファイルと，行毎に各問題の正解画像のファイル名が書かれているテキストファイル，候補画像ファイルが格納されたフォルダの 3 つで構成されている．

trial データの各問題の具体的な詳細は，以下の図 5.1，5.2 の通りである．画像ファイル名が赤く書かれている画像が，その問題の正解画像になる．全フレーズの中の目的単語以外の語句が，その問題の補助語句となっている．

5.1.2　CLIP モデル

CLIP の事前学習済みモデルは，OpenAI が公開しているモデルを用いる．これは，Hugging Face 社の Transformers ライブラリから，モデル名’openai/clip-vit-base-patch32’ で利用できるモデルである．事前学習コーパスには WebImageText が用いられている．このモデルは，画像エンコーダとして ViT-B/32 Transformer アーキテクチャ，テキストエンコーダとして 12 層の Transformer アーキテクチャ の Base モデル，49,152 の語彙数と最大入力長が 76 に設定されている．

5.1.3　WordNet

WordNet は，英語の自然言語のための Python ライブラリである Natural Language Toolkit （NLTK)[*1]に実装されているインターフェースを用いる．NLTK によって提供されている関数を用いることで，WordNet の概念辞書に対して容易に参照することが可能になる．

5.2　実験結果

WordNet にて推定した，各問題の目的単語の定義文と補助語句との類似度は，以下の表 5.1，5.2 の通りである．

CLIP モデルへの入力文は，文字列で入力することが可能である．入力した文章（プロンプト）によって，モデルの出力に影響を与えることがある．よって，与えるプロンプトを工夫することで，CLIP モデルの精度を向上させることが可能になる．

本実験では，与えられた目的単語，全フレーズ，語義定義文の 3 つを用いて，CLIP モデルへのプロンプトの違いによって，精度に影響を与えるかどうかを実験した．

CLIP モデルへのプロンプトは，以下の 3 つに設定した．

[*1] `https://www.nltk.org/index.html`

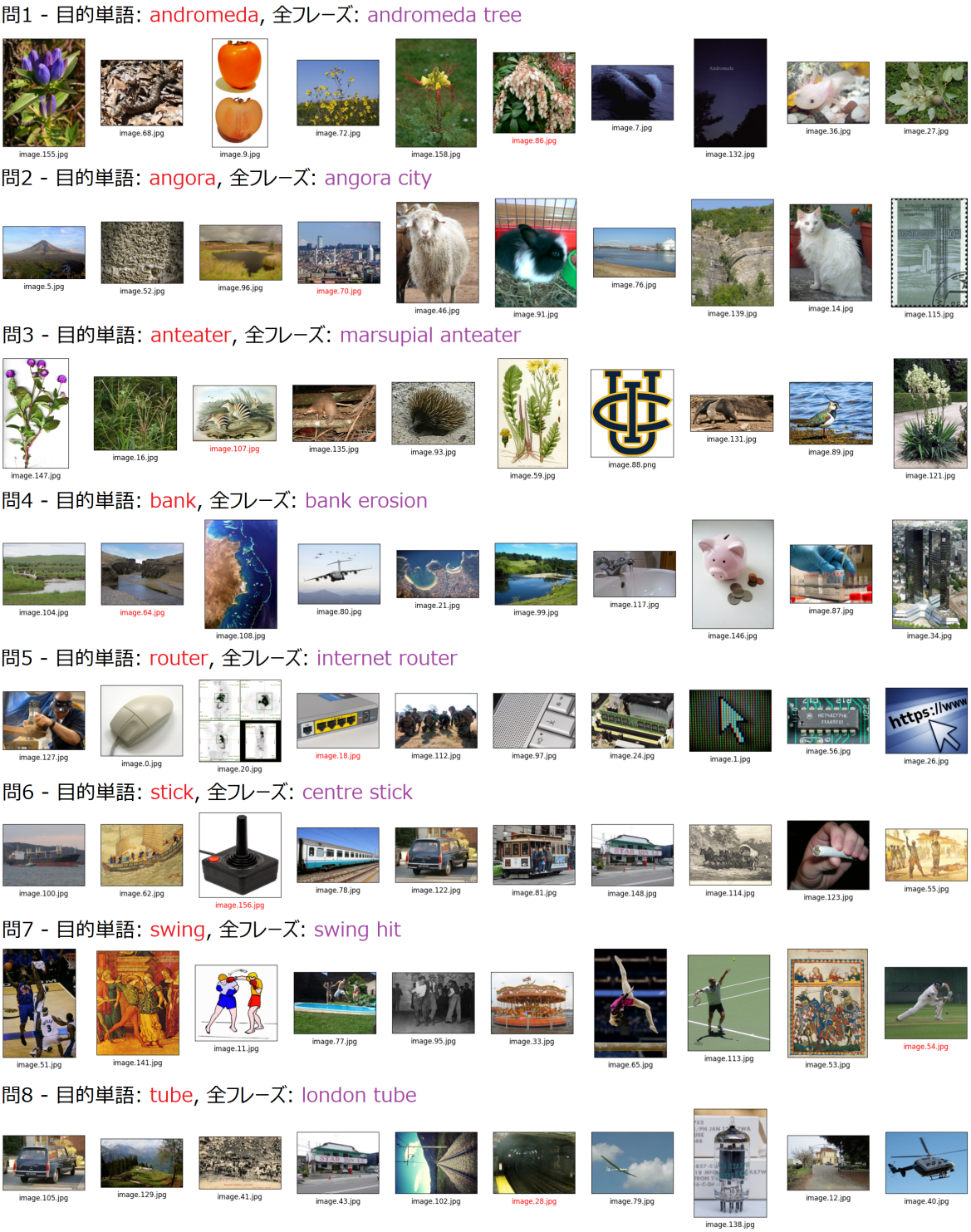


図 5.1: Visual Word Sense Disambiguation タスクの問題例 1

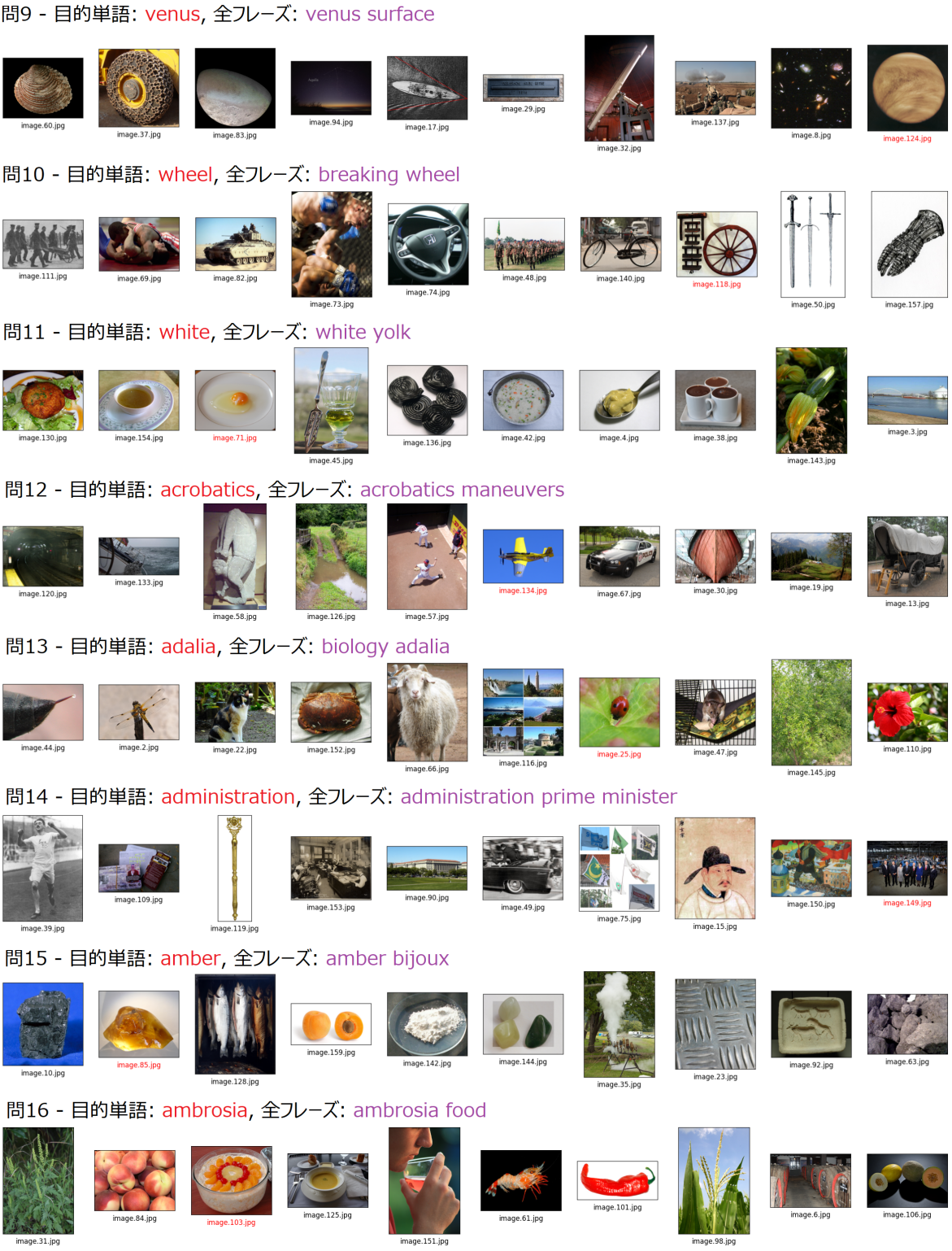


図 5.2: Visual Word Sense Disambiguation タスクの問題例 2

- 目的単語のみ: 'A photo of [目的単語].'
- 全フレーズ: 'A photo of [全フレーズ].'
- 定義文付き: 'A photo of [全フレーズ]. [目的単語] means [語義定義文].'

プロンプトに'A photo of　.' を与えることで，単語だけのプロンプトより精度が向上することが分かっている．これは，CLIP モデルの事前学習の際に，単語だけというケースが珍しく与えるカテゴリーリストが単語だけでは高い精度での予測ができない問題を抱えているからである．また，語義定義文は，目的単語の意味を説明する形式で用いた．

実験結果は，以下の表 5.3 の通りである．また，各プロンプトにおいて，正解できた問題の内訳は，以下の表 5.4 の通りである．

表 5.1: 目的単語の定義文と補助語句との類似度 1

問題	目的単語	定義文	類似度
1	andromeda	broad-leaved evergreen Asiatic shrub with glossy leaves and drooping clusters of white flowers	0.900
2	angora	the capital of Turkey; located in west-central Turkey; it was formerly known as Angora and is the home of Angora goats	0.900
3	anteater	small Australian marsupial having long snout and strong claws for feeding on termites; nearly extinct	0.960
4	bank	sloping land (especially the slope beside a body of water)	0.364
5	router	(computer science) a device that forwards data packets between computer networks	0.706
6	stick	an implement consisting of a length of wood	0.667
7	swing	changing location by moving back and forth	0.778
8	tube	conduit consisting of a long hollow object (usually cylindrical) used to hold and conduct objects or liquids or gases	0.444

表 5.2: 目的単語の定義文と補助語句との類似度 2

問題	目的単語	定義文	類似度
9	venus	the second nearest planet to the sun; it is peculiar in that its rotation is slow and retrograde (in the opposite sense of the Earth and all other planets except Uranus); it is visible from Earth as an early ‘morning star’ or an ‘evening star’	0.533
10	wheel	move along on or as if on wheels or a wheeled vehicle	0.500
11	white	the white part of an egg; the nutritive and protective gelatinous substance surrounding the yolk consisting mainly of albumin dissolved in water	0.933
12	acrobatics	the performance of stunts while in flight in an aircraft	0.800
13	adalia	genus of ladybugs	0.500
14	administration	the persons (or committees or departments etc.) who make up a body for the purpose of administering something	0.143
15	amber	a hard yellowish to brownish translucent fossil resin; used for jewelry	0.235
16	ambrosia	(classical mythology) the food and drink of the gods; mortals who ate it became immortal	0.833

表 5.3: 各プロンプトでの正解率

正解率		
目的単語のみ	全フレーズ	定義文付き
7 / 16 = 0.438	10 / 16 = 0.625	11 / 16 = 0.688

表 5.4: 各プロンプトにおいて正解できた問題の内訳

正答の内訳		
目的単語のみ	全フレーズ	定義文付き
8, 9, 10, 11, 12, 15, 16	2, 5, 6, 8, 9, 10, 11, 12, 13, 16	2, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16

第 6 章

考察

6.1 WordNet による目的単語の語義定義文の推定

NLTK ライブラリに実装されている WordNet の辞書による，目的単語の推定方法による結果は，表 5.1，5.2 に示した通りである．この結果から，V-WSD タスクの各問題の正解画像と比較して，目的単語の語義定義文が間違っている問題は，問題 6，7，8，10，16 の 5 つである．

これらの問題の推論が誤っている要因として考えられるのは，目的単語と補助語句の関係性が，不正解の語義の組合せより類似性が離れているからではないかと考えられる．本実験では，推定アルゴリズムに NLTK ライブラリの WordNet 辞書で実装されている，wup_similarity() 関数を用いている．よって，WordNet のシソーラスから算出できる各々の語義の関係性の計算方法を変更することで，得られる推定結果が変えられることが考えられる．

目的単語の語義定義文が間違っている問題の内，定義文付きのプロンプトで最終的に CLIP モデルが正解できた問題の 3 問（8，10，16）あった．これらの問題が，目的単語の語義定義文が間違っていたのにも関わらず正解することができた要因は，CLIP モデルの事前学習の際に同一の概念を上手く学習できていたからであると考えられる．CLIP モデルは Web 上に数多く存在している多種多様な画像とその説明文のペアから，事前学習を行っているため，V-WSD タスクと似たような，あるいは同一の学習データから知識を得ている可能性は否定できない．

6.2　CLIP モデルによる V-WSD タスクの分析

画像と文章の類似性を計算することが可能な CLIP モデルも利用して V-WSD タスクを行った結果は，表 5.3 の通りである．最も正解率が高かったプロンプトである語義定義文を付与した，

結果より，CLIP モデルに入力するプロンプトは，目的単語のみより補助語句も含むフレーズの方が精度が高くなる．さらに，目的単語の語義定義文を付与することで精度を向上させられることが分かる．

CLIP モデルによる V-WSD タスクの具体的な出力結果は，以下の図 6.1，6.2 の通りである．画像ファイル名の横に書かれている数値は CLIP モデルが出力したものであり，候補画像の集合を分母としたその画像の入力プロンプトに対する類似度を，確率で表したものである．左に行く程，入力プロンプトに類似した画像であり，画像ファイル名が赤字で示された正解画像が，左端にあれば，CLIP モデルによる推論は正解していることになる．

以下では，CLIP モデルによる V-WSD タスクの推論を間違えた要因を，それぞれの問題別に分析する．

6.2.1　語義定義文が正しい問題

WordNet による語義定義文の推定が誤っていた問題以外で，CLIP モデルによる V-WSD タスクの推論結果が間違っていた問題は，問題 1，3，4 の 3 問であった．

問題 1　目的単語「andromeda」は，ツツジ科アセビ属に属する常緑性の低木であるアセビを意味する．この植物は，正解画像（image.86.jpg）が示す通り垂れ下がった白い花の房を持つが，CLIP モデルの出力は，他の小さな白い花を持ち，葉が多く生えている植物の画像の方が類似度が高くなっている．これは，CLIP モデルが「andromeda」や「andromeda tree」という語句の意味と画像との関係性の知識を得ていないためではないかと考えられる．WordNet による目的単語の語義定義文は正しく推定できているが，プロンプト中の知識を得ていない目的単語やフレーズに引っ張られることで，正しい推論ができないと考えられる．実際に，語義定義文のみで CLIP モデルによる推論を行った場合では，正しく推論できていた．

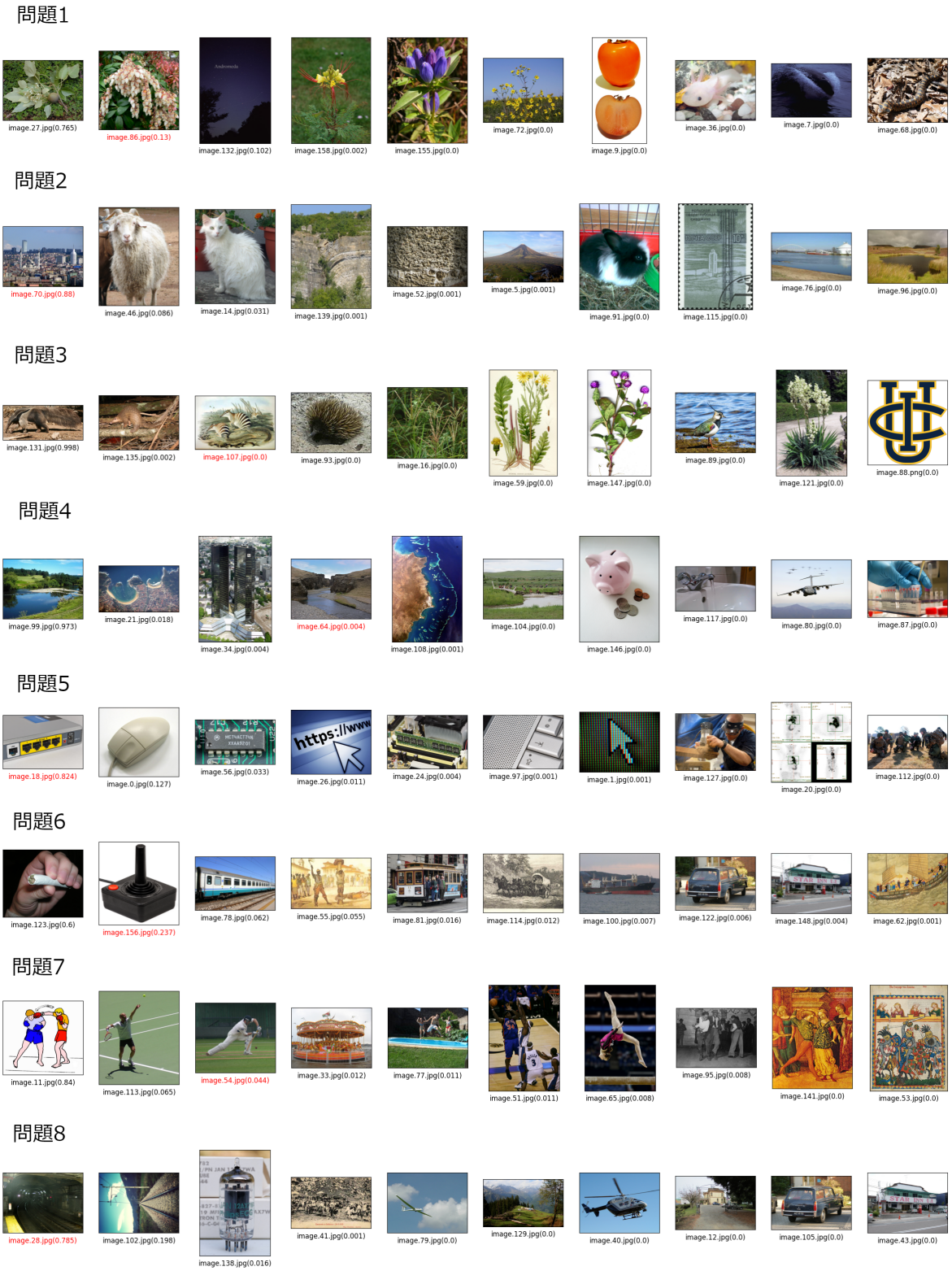


図 6.1: CLIP モデルによる V-WSD タスクの出力例 1

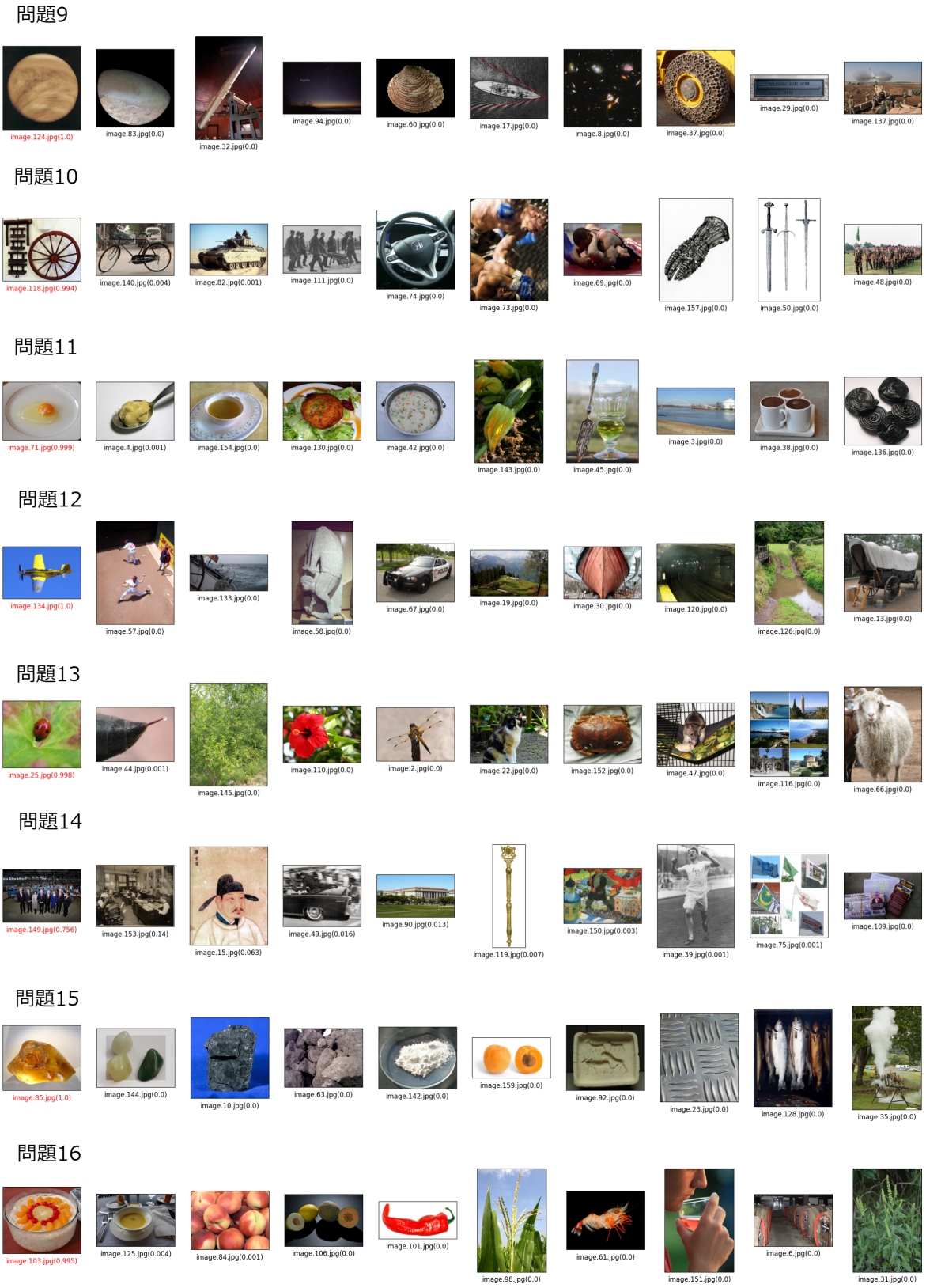


図 6.2: CLIP モデルによる V-WSD タスクの出力例 2

問題 3　目的単語「anteater」は，小型の有袋類で長い鼻とシロアリを食べるための強い爪持つアリクイを意味する．特に，「marsupial anteater」はフクロアリクイと呼ばれる種の学名である．一般名は，「Numbat」と表記される．正解画像（image.107.jpg）は，フクロアリクイのイラストとなっており，CLIP モデルが正しく推論できなかった理由は，イラストの特徴を上手く捉えられなかったためだと考えられる．また，候補画像の中には，一般的なアリクイ（anteater）種の動物が写っている画像（image.131.jpg）も含まれており，「anteater」の意味として一般的でないフクロアリクイであり，フレーズも学名であることを考慮すると，CLIP モデルが推論を間違えてしまう可能性は高いと言える．

問題 4　目的単語「bank」は，特に，水辺の傾斜地を意味する．補助語句「erosion」が付くことで，浸食された川岸を表すが，CLIP モデルは正解画像（image.64.jpg）を正しく推定できていなかった．類似度が最も高かった画像（image.99.jpg）は，正解画像と似たような水辺の傾斜地の画像であり，目的単語の意味にも合致している．この問題は，目的単語より補助語句「erosion」の意味をより強く取ることができれば正しく推論できる可能性がある．

6.2.2　語義定義文が誤っている問題

WordNet による語義定義文の推定が誤っていた問題の中で，CLIP モデルによる V-WSD タスクの推論結果が間違っていた問題は，問題 6，7 の 2 問であった．

問題 6　目的単語「stick」は，パイロットが飛行機の操作をするための操縦桿を意味する．この問題は，間違った語義定義文を付与していることを考慮する必要はあるが，全フレーズのみのプロンプトでは正解できていた問題が，語義定義文を付与したプロンプトでは不正解だった問題である．補助語句が付いた「centre stick」として表記されることが一般的であり，今回与えられたフレーズと一致している．よって，プロンプトに目的単語「stick」の間違った語義定義文を付与して推論させるより，フレーズのみの方が上手く推論できるという結果になった．

問題 7　目的単語「swing」は，特に野球において，打者が投げられたボールを打とうとすることを意味する．補助語句「hit」も，ここでは「swing」と同じような意味を持つ単語であると推論できる．正解画像（image.54.jpg）は，野球競技で打者

がボールをまさに打つところの画像であるが，ボクシングをしているイラスト画像（image.11.jpg）や，テニスでサーブを打とうとしている画像（image.113.jpg）のような，他のスポーツを行っている画像より類似度が低くなってしまっている．このような結果になった要因として，「swing」や「hit」という単語の持つ意味が，スポーツ競技に普遍的に使用される単語であるためだと考えられる．この問題を正しく推論するためには，正解画像が示す競技を，プロンプトに正しく入力する必要があると考えられる．

6.3　CLIP モデルの精度向上

CLIP モデルによる V-WSD タスクの解答を困難にしている要因として，以下の 2 点が挙げられる．

- 目的単語の語義が曖昧であり，候補画像群から正解を絞り込めない
- タスクの精度を向上させるような，CLIP モデルへの入力プロンプトを工夫する必要がある

与えられる目的単語は多義語であり，単語からは候補画像の中から正解を見つけることは非常に困難である．本提案手法では，補助語句を活用することで，WordNet から目的単語の語義を推定する方法を提案しているが，今回使用したアルゴリズムでは，16 問中 5 問の語義推定が誤っている．この部分のアルゴリズムを，より精錬することで語義推定の誤り率を下げることができる余地がある．

また，WordNet による目的単語の語義推定が正確に行えたとしても，CLIP モデルに入力する際に，プロンプトに工夫を施す必要があることが判明している．全ての問題に対して，普遍的に使用することができるプロンプトを調査することで，CLIP モデルの精度を向上させられるといえる．

第 7 章

結論

本研究では，与えられた語義が曖昧な単語とその補助語句を用いて，候補画像の中から正解の画像を選択するタスクである V-WSD に対して，入力した画像と言語の類似性を計算することが可能な事前学習モデルである CLIP を活用してタスクの解答を行い，精度向上の手法について提案した．

提案手法では，WordNet を用いて多義語である目的単語の語義を推定できるアルゴリズムを実装し，CLIP モデルのプロンプトに活用できる語義定義文を推定した．推定した語義定義文をプロンプトに取り込み，全ての問題に対して普遍的に有効である文章になるように定義した．

実験では，WordNet による語義推定の結果から，各問題毎の CLIP モデルの出力に対して評価を行い，精度向上のための分析を行った．WordNet を利用した CLIP モデルによる V-WSD の精度より，提案手法が，Vision-and-Language 分野のタスクの 1 つである V-WSD において，一定の有用性を持つことを示した．

今後の課題として，WordNet による語義推定アルゴリズムは一定の精度を発揮したが，改善の余地があることが分かった．各単語の語義の関係性を計算するアルゴリズムには，幾つかの手法が存在しているため，それらを試すことで精度が向上するか調査を行う．また，CLIP モデルへのプロンプトの工夫も精度向上への余地があるといえる．全ての問題に対して普遍的に使用できるプロンプトを開発することで，精度向上が期待できるといえる．

謝辞

本研究を進めるにあたって，多くのご指導を頂いた指導教員の新納浩幸教授に感謝致します．また，日常の議論を通して多くの知識，示唆を頂いた新納研究室の皆様に感謝します．

参考文献

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

[2] Mayumi Hiyoshi and Hideo Shimazu. Drawing pictures with natural language and direct manipulation. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*, 1994.

[3] Naoyuki Okada. Conceptual taxonomy of Japanese verbs for uderstanding natural language and picture patterns. In *COLING 1980 Volume 1: The 8th International Conference on Computational Linguistics*, 1980.

[4] Terry Winograd. Understanding natural language. *Cognitive psychology*, Vol. 3, No. 1, pp. 1–191, 1972.

[5] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 37, No. 11, pp. 2332–2345, 2015.

[6] Haofeng Zhang, Yang Long, Yu Guan, and Ling Shao. Triple verification network for generalized zero-shot learning. *IEEE Transactions on Image Processing*, Vol. 28, No. 1, pp. 506–517, 2018.

[7] Guanyu Yang, Zihan Ye, Rui Zhang, and Kaizhu Huang. A comprehensive survey of zero-shot image classification: methods, implementation, and fair evaluation. *Applied Computing and Intelligence*, Vol. 2, No. 1, pp. 1–31, 2022.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, Vol. 63, No. 11, pp. 139–144, 2020.

[10] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: a survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 45, No. 1, pp. 539–559, 2022.

[11] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11162–11173, 2021.

[12] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pp. 153–170. Springer, 2020.

[13] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

付録

A　提案手法の実験のために使用したソースコード

提案手法の手順 3 から 5 で，問題の目的単語と補助語句から，2 つの語句の類似度を計算し，推定した目的単語の語義定義文を CLIP モデルへと入力して，V-WSD タスクの正解率を出力するプログラムのソースコードを A.1 に示す.

ソースコード A.1: v-wsd.py

```
import warnings
warnings.simplefilter('ignore')

import re
import pickle
from tqdm import tqdm
from PIL import Image, ImageFile
import torch
from torchvision import transforms
from transformers import CLIPProcessor, CLIPModel
from nltk.corpus import wordnet as wn

Image.MAX_IMAGE_PIXELS = None
ImageFile.LOAD_TRUNCATED_IMAGES = True
torch.set_printoptions(sci_mode=False)

device = "cuda:0" if torch.cuda.is_available() else "cpu"
transform = transforms.Resize(600, interpolation=Image.BICUBIC)

PATH = './semeval-2023-task-1-V-WSD-v1/trial_v1'
data_txt = f'{PATH}/trial.data.v1.txt'
gold_txt = f'{PATH}/trial.gold.v1.txt'
imgpath = f'{PATH}/trial_images_v1/'
```

```
24
25 pklpath = f'captions.pkl'
26
27 clipmodel = CLIPModel.from_pretrained("openai/clip-vit-base-patch32")
28 clipmodel = clipmodel.to(device)
29 processor = CLIPProcessor.from_pretrained("openai/clip-vit-base-patch32
       ")
30
31 datalist = []
32 with open(data_txt, encoding="utf-8") as dataf:
33     with open(gold_txt, encoding="utf-8") as goldf:
34         for line, gold in zip(dataf, goldf):
35             linedict = {}
36             linelist = re.split('\t', line.rstrip())
37             linedict['target_word'] = linelist[0]
38             linedict['full_phrase'] = linelist[1]
39             linedict['imgs'] = linelist[2:]
40             linedict['gold'] = [gold.rstrip('\n'), linedict['imgs'].
                   index(gold.rstrip('\n'))]
41             datalist.append(linedict)
42
43 with open(pklpath, 'rb') as tf:
44     sents_dic = pickle.load(tf)
45
46 for problem in datalist:
47     texts = problem['full_phrase'].split()
48     sup_word = '_'.join([s for s in texts if not s == problem['
           target_word']])
49     targets = wn.synsets(problem['target_word'])
50     sup_words = wn.synsets(sup_word)
51     comp = 0
52     for syn_t in targets:
53         if len(syn_t.hypernyms()) == 1:
54             syn_comp = syn_t.hypernyms()[0]
55         else:
56             syn_comp = syn_t
57         for syn_s in sup_words:
58             similarity = syn_comp.wup_similarity(syn_s)
59             if comp < similarity:
60                 comp = similarity
61                 target_def = syn_t.definition()
```

```
62
63      problem['definition'] = target_def
64
65  iterFor = tqdm(datalist)
66  reslist = []
67  for i, dic in enumerate(iterFor):
68      imgs = [transform(Image.open(imgpath + img)) for img in dic['imgs'
            ]]
69      txt_word = f"A␣photo␣of␣{dic['target_word']}."
70      txt_full = f"A␣photo␣of␣{dic['full_phrase']}."
71      txt_def = f"A␣photo␣of␣{dic['full_phrase']},␣{dic['target_word']}␣
            means␣{dic['definition']}."
72
73      with torch.no_grad():
74          inputs_word = processor(text=txt_word, images=imgs,
                return_tensors="pt", padding=True).to(device)
75          outputs_word = clipmodel(**inputs_word)
76          probs_word = outputs_word.logits_per_text.softmax(dim=1).to('
                cpu')
77          pred_class_idx_word = probs_word.argmax(-1).item()
78          dic['probs_word'] = probs_word.squeeze()
79          dic['pred_cls_word'] = pred_class_idx_word
80
81          inputs_full = processor(text=txt_full, images=imgs,
                return_tensors="pt", padding=True).to(device)
82          outputs_full = clipmodel(**inputs_full)
83          probs_full = outputs_full.logits_per_text.softmax(dim=1).to('
                cpu')
84          pred_class_idx_full = probs_full.argmax(-1).item()
85          dic['probs_full'] = probs_full.squeeze()
86          dic['pred_cls_full'] = pred_class_idx_full
87
88          inputs_def = processor(text=txt_def, images=imgs,
                return_tensors="pt", padding=True).to(device)
89          outputs_def = clipmodel(**inputs_def)
90          probs_def = outputs_def.logits_per_text.softmax(dim=1).to('cpu
                ')
91          pred_class_idx_def = probs_def.argmax(-1).item()
92          dic['probs_def'] = probs_def.squeeze()
93          dic['pred_cls_def'] = pred_class_idx_def
94
```

```
95      reslist.append(dic)
96
97  ok_word = 0; ok_full = 0; ok_def = 0
98  ok_word_l = []; ok_full_l = []; ok_def_l = []
99  for i, dic in enumerate(reslist):
100     if dic['pred_cls_word'] == dic['gold'][1]:
101         ok_word += 1; ok_word_l.append(i+1)
102     if dic['pred_cls_full'] == dic['gold'][1]:
103         ok_full += 1; ok_full_l.append(i+1)
104     if dic['pred_cls_def'] == dic['gold'][1]:
105         ok_def += 1; ok_def_l.append(i+1)
106
107 print(f'Acc(target_word):␣{ok_word}␣/␣{len(reslist)}␣=␣{ok_word/len(
        reslist):.3f}')
108 print(f'Acc(target_word)␣List:␣{ok_word_l}')
109 print(f'Acc(full_phrase):␣{ok_full}␣/␣{len(reslist)}␣=␣{ok_full/len(
        reslist):.3f}')
110 print(f'Acc(full_phrase)␣List:␣{ok_full_l}')
111 print(f'Acc(definition):␣{ok_def}␣/␣{len(reslist)}␣=␣{ok_def/len(
        reslist):.3f}')
112 print(f'Acc(definition)␣List:␣{ok_def_l}')
```