論文 / 著書情報
Article / Book Information

論題(和文)	事前学習済みモデルを用いた日本語直喩表現の解釈
Title(English)	
著者(和文)	鈴木颯仁, 山田寛章, 徳永健伸
Authors(English)	Hayato Suzuki, Hiroaki Yamada, Takenobu Tokunaga
出典(和文)	言語処理学会第30回年次大会 発表論文集, pp. 3137-3142
Citation(English)	Proceedings of the Thirtieth Annual Meeting of the Association for Natural Language Processing, pp. 3137-3142
発行日 / Pub. date	2024, 3
権利情報 / Copyright	(C)言理処理学会，クリエイティブ・コモンズ・ライセンス（表示 4.0国際）https://creativecommons.org/licenses/by/4.0/

事前学習済みモデルを用いた日本語直喩表現の解釈

鈴木颯仁　山田寛章　徳永健伸
東京工業大学 情報理工学院
{suzuki.h.ci@m, yamada@c, take@c}.titech.ac.jp

概要

直喩表現（例：ひまわりのような笑顔）に対して，人のような自然な解釈（例：明るい笑顔）の候補を生成するモデルを作成することは，自然言語処理の分野において注目を集めている課題のひとつである．本研究では，事前学習済みマスク言語モデル BERT[1] を用いて直喩表現に対する解釈を生成する．また，形容詞の補完に適したマスク言語モデル (Masked Language Model，MLM) の拡張手法と形容詞-名詞の修飾関係に着目した学習フレームワークを提案する．提案手法の適用によって，直喩解釈のスコアを表す Recall@5 は 0.296 を示し，他比較対象を上回った．

1　背景

比喩表現はある物事を他の物事に喩える表現手法であり，日常の言語使用から科学的な言説に至るまで広く用いられる．比喩表現の一種である直喩表現は「S のような T」という形で，T を S に喩えて伝える．このとき，喩える語 S を喩辞，喩えられる語 T を被喩辞と呼ぶ．比喩表現に対する解釈を生成することは，自然言語処理における重要な課題である．例えば，「ひまわりのような笑顔」という表現を字義通りに「ひまわり」と「笑顔」を同等のものとして解釈するのか，比喩表現であると理解した上で「ひまわりのように “明るい” 笑顔」と解釈するのかでは大きな差がある．この差異を捉えることは，自然言語処理の重要なタスクである感情分析 [2]，機械翻訳 [3], 質問応答 [4] で不可欠なものである．

直喩解釈データセットを作成するためには各直喩表現に対して複数人による解釈付けを行う必要があり，コストが高いため，現在使用可能なデータの量は限られている．このため，一定量の直喩解釈データを必要とするファインチューニングが必要な事前学習済みモデル (Pre-trained Language Model, PLM) による直喩表現解釈生成の研究は十分検討されていない．大村 [5] の研究では，word2vec による単語埋め込みと共起情報を活用して直喩表現の解釈を生成している．日本語以外の直喩解釈について PLM を用いた研究として Weijie ら [6] の研究が挙げられる．この研究では，PLM に形容詞と名詞の関係を学習させた後，最適なプロンプトを探索することでスコアの向上を図っている．

表 1　形容詞をトークナイザー [1)] で分割した時のトークン数の分布

トークン数	割合 (%)
1	1.7
2	23.3
3	55.3
4	17.6
>4	2.1

LM で広く採用されているサブワード分割を適用すると，日本語の形容詞の多くは表 1 に示すとおり複数トークンに分かれてしまう．しかし，BERT などマスク言語モデル (Masked Language Model，MLM) では単一トークンの補完を目的として学習を行っているため，複数トークンの補完を行うことは難しい．複数トークンの補完を可能にする研究として，Cui ら [7] の研究がある．この研究では，一つの単語から分割された複数トークン全てに対してマスク処理を行い，学習させる whole-word-masking という手法を用いている．しかし，この手法には補完した単語の尤度を正確に計算することができず，またマスク処理を行うトークンの数を事前に決める必要があるという問題がある．Oren ら [8] の研究ではこれらの問題を解決するために，MLM のデコーダー部分の行列を拡張することで複数トークンの補完を単一トークンの補完へ帰着させている．

本研究では，「S のような T」(例: ひまわりのような笑顔) という直喩表現が与えられたとき，解釈となる形容詞の集合 (例: 明るい，暖かい) を生成す

1) https://huggingface.co/cl-tohoku/bert-base-japanese-v3

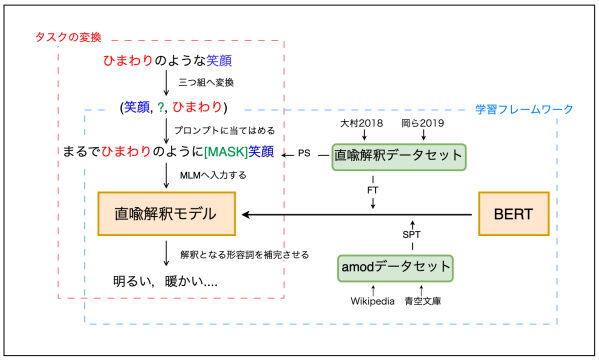

図 1　学習フレームワークとタスクの変換

る問題を扱う．また，直喩表現を (被喩辞, 解釈, 喩辞) の 3 つ組として表現する [9] ことで，直喩表現の解釈を生成するタスクを MLM のタスクに帰着し (図 1)，MLM を用いた直喩解釈を可能にする．さらに，単一マスクで日本語形容詞の補完を可能にする MLM の拡張手法と直喩解釈に関する知識を PLM に学習させる学習フレームワークを提案する．

2　データセット

本研究では，amod[2] データセットと直喩解釈データセットの二つのデータセットと，モデルの補完可能語彙となる形容詞の集合を作成する．

2.1　amod データセット

青空文庫コーパス [10] と Wikipedia 日本語コーパスに含まれる文章に対して GiNZA[11] を用いて係り受け解析を行い，amod 関係を含む文を抽出する．その後，抽出した文中で amod 関係にある形容詞に対してマスク処理を行う．ただし，amod 関係にあるとして抽出された形容詞のうち，出現頻度の少ないものは形容詞として相応しくないことが多いため，形容詞は抽出された文中に 20 回以上出現するもののみに絞る．さらに，各形容詞の出現回数の差を少なくするために，50 回以上マスク処理されないようにする．このようにして，学習用：167,006，検証用：20,876，テスト用：20,876 のデータセットを作成した．付録表 6 に具体例を示す．

2.2　直喩解釈データセット

大村 [5] と岡ら [12] の研究で作成された直喩解釈データセットをもとに作成する．これらのデータセットには，喩辞と被喩辞の組に対して解釈となる形容詞が人手により複数個付与されている．ただし，岡らの研究で付与された解釈には形容詞以外の単語が含まれていたため人手で除外する．このようにして，学習用：239，検証用：119，テスト用：120 のデータセットを作成した．付録表 7 に具体例を示す．

2.3　補完可能語彙となる形容詞の集合

補完可能語彙となる形容詞の集合は前記の amod データセット（2.1 節）に含まれる形容詞を用いており，4,830 種類の形容詞が含まれる．この形容詞の集合は直喩解釈データセット（2.2 節）に含まれる解釈となる形容詞のうち 97.51%をカバーしている．

3　提案手法

本研究では，単一マスクで日本語形容詞の補完を可能にする MLM の拡張手法である Adjective-Specific Matrix (ASMT) Decoder と，直喩解釈に関する知識を PLM に学習させるための学習フレームワークを提案する．

3.1　ASMT Decoder

Oren ら [8] の研究を参考に，MLM のデコーダー部分の行列に含まれる語彙に元は複数トークンから構成される形容詞を追加することで，単一マスクによる形容詞の補完を可能にした．さらに，本研究で補完の対象としている単語は形容詞のみであるため，行列に含まれる語彙を形容詞に限定した．

本研究では形容詞の集合（2.3 節）を補完可能な語彙とする ASMT Decoder を作成することで，単一マスクで補完可能な形容詞が 82 種類から 4,380 種類に増加した．

3.2　学習フレームワーク

直喩解釈に関する知識を PLM に学習させるために，以下の 3 段階に分かれた学習フレームワークを提案する (図 1)．

3.2.1　Secondary Pre-Training (SPT)

SPT では ASMT Decoder を学習させること，より豊富な形容詞を出力させることを目的として，amod データセットを用いて追加事前学習を行う．

MLM の事前学習時にはクロスエントロピーロスを最小化するため，学習時に出現頻度の高い単語が出力されやすい [13]．しかし，直喩表現の解釈で出

2)　amod 関係とは名詞（代名詞）とそれを修飾する形容詞の関係のことを指す．例えば「彼は酷い怪我をした．」という文では，「酷い」と「怪我」が amod 関係にある．

表 2　作成したプロンプトの例

	プロンプト	例
p_1	まるで (喩辞) のように (解釈) (被喩辞)	まるでひまわりのように明るい笑顔
p_2	とても (解釈) (喩辞)	とても明るいひまわり
p_3	とても (解釈) (被喩辞)	とても明るい笑顔

現する形容詞の分布は MLM の事前学習時に出現する形容詞の分布と異なる．SPT で用いる amod データセットは各形容詞の出現回数の差を減らしているため，高頻度で出現する単語のみに偏って出力するバイアスを軽減することが期待できる．

3.2.2　Fine-Tuning (FT)

FT では直喩表現解釈の知識をモデルに学習させるために，直喩解釈学習データを用いてファインチューニングを行う．ただし，複数ある解釈を分割し，複数個の（被喩辞，解釈，喩辞）の三つ組に変換して用いる．

3.2.3　Prompt Search (PS)

PS では直喩解釈に適するプロンプトを見つけることを目的として，直喩解釈検証データを用いてプロンプトの探索を行う．ただし，プロンプトとは直喩表現の三つ組を当てはめるためのプレースホルダーを持つ文字列を指す．

LM はプロンプトに敏感であるため，引き出せる知識はプロンプトに大きく影響される．さらに，一つのプロンプトだけではなく複数のプロンプトを用いることで精度の向上を見込める [14]．したがって，本研究では基本となるプロンプト（p_1）の他に，喩辞に注目したプロンプト（p_2），被喩辞に注目したプロンプト（p_3）を作成し（表 2），これらのプロンプトを組み合わせることで，さらなるスコアの向上を目指す．Ortony[15] の研究によると，比喩表現によって強調された属性は喩辞と被喩辞が共有する属性であるが，喩辞の方がより強くその属性を表すため，p_2 は p_3 よりも有効であると考えられる．

あるプロンプトの組み合わせ P に対する出力分布 Q_P は以下のように計算される．

$$Q_P(w|P) = \frac{1}{|P|}\Sigma_{p\in P}\log(Q(w|p)) \tag{1}$$

このように計算された出力分布をもとに解釈を生成して，スコアを計算し，最良となるプロンプトの組み合わせを探す．

4　実験

直喩解釈実験に加えて，SPT の有効性を単体で評価する SPT 実験を行う．

4.1　実験設定

BERT として cl-tohoku/bert-base-Japanese-v3 を用いる．SPT は amod データセットを用いて行い，最適化アルゴリズムを Adam，学習率を 5e-5，バッチサイズを 128 としている．FT は直喩解釈データセットを用いて行い，最適化アルゴリズムを Adam，学習率を 5e-5，バッチサイズを 64 とし，Decoder 部分のみを学習させる．また，双方とも Early Stopping を用いる．PS では，直喩解釈検証データを用いた評価の結果，スコアが最も高かったプロンプトの組み合わせ $\{p_1,p_2\}$ を用いる．

4.2　評価指標

SPT 実験では，形容詞補完の精度を測るために Top-K Accuracy (a@K)，出力する形容詞の多様さを測るために式 (2) で定義される Type@K を用いた．

$$\text{Type@K} = \frac{\text{Num(Unique\_words)}}{K*N} \tag{2}$$

ただし，Num(Unique_words) は出力された形容詞の種類，N は amod テストデータのサイズを表す．

直喩解釈の性能評価指標としては，直喩解釈の研究で用いられることの多い Recall@K（r@K）を用いた．

4.3　比較対象

SPT 実験　BERT に ASMT Decoder を適用した BERT-ASMT をベースラインとして，これに SPT を適用した BERT-ASMT + SPT と比較する．

直喩解釈実験　以下の 4 手法をベースラインとする．(1) 大村 2018[5]：word2vec による類似度とバイグラムによる共起頻度を元に形容詞の集合の中から直喩表現の解釈として尤もらしい単語を選ぶ．ただし，word2vec は jawiki.all_vectors.200d.txt[3] を用いる．テストデータのうち 14 個は喩辞または被喩辞が word2vec の語彙に含まない単語であるため，解釈を生成できない．(2) BERT：提案手法を一切適用しない BERT．(3) ChatGPT Zero-shot：モデルは GPT-3.5-turbo[16] を用い，その他設定については付録表 8 の通りである．(4) ChatGPT One-shot：(3) に事

3)　https://github.com/singletongue/WikiEntVec?tab=readme-ov-file

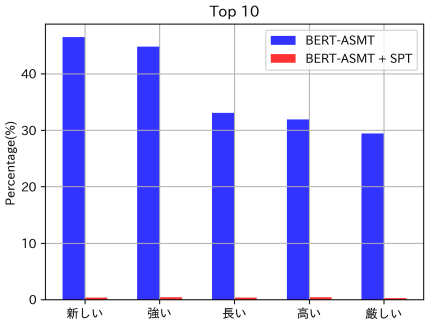

図 2　Top10 に各形容詞が含まれている頻度

表 3　Top-K Accuracy

手法	a@5	a@10	a@15	a@20
BERT-ASMT	0.015	0.016	0.017	0.018
BERT-ASMT + SPT	0.397	0.482	0.533	0.568

表 4　Type@K

手法	t@5	t@10	t@15	t@20
BERT-ASMT	0.017	0.015	0.012	0.010
BERT-ASMT + SPT	0.046	0.023	0.015	0.012

表 5　Recall@K

手法	r@5	r@10	r@15	r@20
(1) 大村 2018	0.068	0.127	0.175	0.200
(2) BERT	0.089	0.145	0.183	0.204
(3) ChatGPT Zero-shot	0.160	0.188	0.235	0.240
(4) ChatGPT One-shot	0.294	0.332	0.347	0.348
(a) ASMT	0.203	0.266	0.314	0.338
(b) ASMT + SPT	0.090	0.133	0.174	0.196
(c) ASMT + FT	0.251	0.360	0.431	0.466
(d) ASMT + SPT + FT	0.286	0.402	0.472	0.527
(e) ASMT + SPT + FT + PS	**0.296**	**0.438**	**0.504**	**0.557**

例を一つ与える．与える事例は直喩解釈学習データからランダムで 5 個選んだうち最もスコアが高いものとする．さらに，提案手法に対するアブレーション比較対象として以下を用いる．(a) ASMT：BERT に ASMT Decoder を適用する．(b) ASMT+SPT：(a) に SPT を適用する．(c) ASMT+FT：(a) に FT を適用する．(d) ASMT+SPT+FT：(a) に SPT と FT を適用する．(e) ASMT+SPT+FT+PS：(a) に SPT，FT，PS を適用する．各手法は 5 回実験し，平均スコアを報告する．

5　結果と考察

5.1　SPT の有効性

a@K と t@K をそれぞれ表 3，表 4 に示す．a@K と t@K は，学習前よりも向上している．また，amod テストデータにおいて，ある形容詞がモデル出力の上位 10 個中に含まれる頻度を計算し，SPT 適用前後で比較した．図 2 に SPT 適用前に出力頻度が高かった上位 5 個の形容詞について，SPT 適用前後での頻度の変化を示す．SPT 適用前に出力頻度が高い形容詞の上位 5 個のいずれにおいても，適用後に出力頻度が大きく減少していることがわかる．以上から，SPT によって，多様な形容詞を補完対象として，解釈となる形容詞を適切に出力できるモデルを作成できると考えられる．

5.2　直喩解釈性能

直喩解釈の結果を表 5 に示す．太字は最も良いスコアを示している．FT を適用したモデル (c,d,e) は，適用しないモデル (a,b) に比べてスコアが高い．この結果より，FT によってモデルは直喩解釈の知識を得ることができたと考えられる．また，FT のみを適用した (c) よりも，SPT と FT を適用したモデル (d) の方がスコアが高い．これは，直喩解釈学習データに出現しない形容詞の知識を，SPT を通して学習できたことが要因であると考えられる．しかし，SPT のみを行ったモデル (b) はスコアが下がった．これは，SPT で行っているタスクと直喩表現を解釈するタスクの差が大きいことが原因であると考えられる．直喩解釈検証データを用いた PS の結果，p_3 よりも p_2 を用いた方がスコアが高くなっている（PS の結果は付録表 9 参照）．この結果から，被喩辞よりも喩辞から適切な解釈を生成しやすいと考えられ，Ortony[15] による主張を裏付ている．

6　結論

本研究では，事前学習済みマスク言語モデルである BERT を用いて直喩解釈を行うためのモデルの拡張手法，そして学習フレームワークを提案した．

今後の課題としては次の二つが考えられる．一つ目は，SPT タスクをより直喩解釈タスクに近づけることである．SPT タスクと直喩解釈タスクの違いが大きいため SPT 前後でスコアが低下している．二つ目は，出力対象の拡張である．本研究では直喩表現解釈として形容詞のみを扱ったが，実際には形容詞節等も解釈として存在しうる．より一般的な直喩表現に対して解釈を生成するためには，形容詞節等も出力できるよう拡張する必要がある．

謝辞

本研究で使用した直喩解釈データの一部を提供していただいた岡隆之介氏に感謝申し上げます.

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, Vol. 1, pp. 4171–4186, 2019.

[2] Rui Mao, Xiao Li, Mengshi Ge, and Erik Cambria. Metapro: A computational metaphor processing model for text pre-processing. **Information Fusion**, Vol. 86–87, pp. 30–43, 2022.

[3] Rui Mao, Chenghua Lin, and Frank Guerin. Word embedding and wordnet based metaphor identification and interpretation. **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics**, Vol. 1, pp. 1221–1231, 2018.

[4] Wei Zhao, Haiyun Peng, Steffen Eger, Erik Cambria, and Min Yang. Towards scalable and reliable capsule networks for challenging nlp applications. **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 1549–1559, 2019.

[5] 大村雅一. 分散表現を用いた比喩表現の解釈, 2018.

[6] Weijie Chen, Yongzhu Chang, Rongsheng Zhang, Jiashu Pu, Guandan Chen, Yadong Xi Le Zhang, and Chang Su Yijiang Chen. Probing simile knowledge from pre-trained language models. **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics**, Vol. 1, pp. 5875–5887, 2022.

[7] Yiming Cui, Wanxiang Che, Ting Liua, Ziqing Yang Bing Qin, Shijin Wang, and Guoping Hu. Pre-training with whole word masking for chinese bert. **IEEE/ACM Transactions on Audio, Speech and Language Processing**, pp. 3504–3514, 2021.

[8] Oren Kalinsky, Guy Kushilevitz, Alexander Libov, and Yoav Goldberg. Simple and effective multi-token completion from masked language models. **Findings of the Association for Computational Linguistics: EACL 2023**, pp. 2356–2369, 2023.

[9] Wei Song, Jingjin Guo, Ruiji Fu, Ting Liu, , and Lizhen Liu. A knowledge graph embedding approach for metaphor processing. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, Vol. 29, pp. 406–420, 2020.

[10] 青空文庫, (2024-1 閲覧). https://www.aozora.gr.jp.

[11] GiNZA - Japanese NLP Library, (2024-1 閲覧). https://megagonlabs.github.io/ginza/.

[12] 岡隆之介, 大島裕明, 楠見孝. 比喩研究のための直喩刺激-解釈セット作成および妥当性の検討. 心理学研究, Vol. 90, No. 1, pp. 53–62, 2019.

[13] Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. Gltr: Statistical detection and visualization of generated text. **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations**, pp. 111–116, 2019.

[14] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. **Transactions of the Association for Computational Linguistics**, Vol. 9, pp. 962–977, 2021.

[15] Andrew Ortony. Beyond literal similarity. **Psychological Review**, Vol. 86, pp. 160–181, 1979.

[16] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models, 2023.

付録

データセットの例

表 6 amod データセットの例

形容詞	名詞	文章
酷い	ケガ	彼はとても [MASK] ケガをしている
白い	鳥	湖に [MASK] 鳥がいる

表 7 直喩解釈データセットの例

喩辞	被喩辞	解釈
鬼	表情	怖い, 恐ろしい, 険しい...
子供	顔	幼い, 可愛い, 無邪気な...

ChatGPT における設定

ChatGPT における temperature は 0 と設定した．表 8 に用いたプロンプトを示す．その他記載のない設定はデフォルト値である．

表 8 ChatGPT で用いたプロンプト

種別	プロンプト内容
システムプロンプト	出力は以下の形式を守って，K 個出力してください． ・出力 1 ・出力 2 ⋮
ユーザプロンプト	次の直喩表現の解釈の候補となる形容詞を K 個出力してください． ただし，形容詞は連体形で出力してください． S のような T

Prompt Search の詳細

直喩解釈検証データを用いて Prompt Search を行った結果を示す．

表 9 Prompt Search の結果

プロンプト	r@5	r@10	r@15	r@20
$\{p_1\}$	0.288	0.404	0.490	0.533
$\{p_2\}$	0.249	0.353	0.416	0.469
$\{p_3\}$	0.143	0.226	0.288	0.332
$\{p_1, p_2\}$	**0.318**	**0.436**	**0.494**	**0.545**
$\{p_2, p_3\}$	0.245	0.355	0.425	0.472
$\{p_3, p_1\}$	0.257	0.362	0.438	0.486
$\{p_1, p_2, p_3\}$	0.294	0.406	0.486	0.538