一般性を考慮した言語処理モデルの Shortcut Reasoning の自動検出

原口大地 [1]　白井清昭 [1]　井之上直也 [1,2]
[1] 北陸先端科学技術大学院大学　[2] 理化学研究所
{s2110137,kshirai,naoya-i}@jaist.ac.jp

概要

Shortcut Reasoning は言語処理モデルの非合理的な推論であり，頑健性を損ねる主な原因とされている．先行研究では，Shortcut Reasoning を発見し，低減させる試みが行われているが，それらの手法は様々な問題点を抱えている．本研究では，推論パターンとその一般性をモデルから抽出・計算し，Shortcut Reasoning を自動的に検出する手法を提案する．実験の結果，提案手法は既に明らかになっている Shortcut Reasoning を検出できたことに加え，未知のものを発見することにも成功した．

1　はじめに

近年，事前学習モデルをはじめとした自然言語処理モデルがあらゆるタスクで精度の向上を見せている．一方で，学習データ内の交絡あるいは疑似相関 (Spurious Features) [1, 2, 3] にモデルが依存することで発生する，推論プロセスにおける非合理的な推論 (**Shortcut Reasoning**) が指摘されている [4, 5, 6].

Shortcut Reasoning は，学習データと同じ分布を持つデータ (Independent and Identically Distributed: IID) と比べて，異なる分布を持つデータ (Out of Distribution: OOD) の解析性能を下げる，すなわち頑健性を低下させることが懸念されている．

Shortcut Reasoning を発見・解消させようとする試みは既に行われているが [7, 8, 9]，そこで提案された手法は，(i) Shortcut Reasoning の形態について事前に想定している，(ii) モデルの内部情報を考慮していない，(iii) 人手による判定を必要としている という制約を持っている．(i) は事前に想定した Shortcut Reasoning に対してのみ検証をするため，モデルに潜在する未知のものを明らかにできない可能性がある．(ii) については，内部情報を使わない手法の多くが何らかの編集や特徴を加えた入力を作成し，それに対する出力を分析することで，Shortcut Reasoning の存在を明らかにしようとしているが，この手法によって我々が知り得るのはモデルの出力のみであり，得られる情報には限度がある．(iii) に関しては，単純にコストがかかるのに加え，一見 Shortcut Reasoning に見えないような事例を見逃す可能性がある等の課題を抱えている．

以上の 3 つの課題の解決に取り組んだ研究 [10] はいまだ少数であり，十分な研究成果が得られていないのが現状である．本論文は，最小限の仮定で，内部情報を利用しながら，人手での判定を必要とせずに自動的に Shortcut Reasoning を検出する手法を提案する．具体的には，モデルの推論プロセスにおける規則性 (**推論パターン**) を抽出し，IID と OOD の入力に対する有効性を比較することで，Shortcut Reasoning の存在とその形態を明らかにする．

2　Shortcut Reasoning の検出

本節では，Shortcut Reasoning の検出に先立ち，推論パターン (2.1 項) とその抽出方法 (2.2 項)，一般性 (2.3 項)，Shortcut Reasoning の判定手法 (2.4 項) を定義し，これを踏まえて Shortcut Reasoning 検出の具体的な手順を説明する (2.5 項).

2.1　推論パターン

本研究では，ある入力に対するモデル f の推論プロセスにおいて，何らかのトリガー (Trigger) が特定のラベル (Label) の予測をもたらす規則を推論パターンと定義する．推論パターン p は形式的に次のように定義できる．

$$p \stackrel{\text{def}}{=} \text{Trigger} \xrightarrow{f} \text{Label} \tag{1}$$

以降，簡易な記法として $p = (t, l)$ と表す．t はトリガー，l はラベルを表す．

2.1.1 推論パターンの分類

Shortcut Reasoning は学習データの Spurious Features によりもたらされるが，それらが推論パターンにおけるトリガーとラベルであることは，その推論パターンが Shortcut Reasoning であることの必要条件といえる．したがって，推論パターンの定義において，Spurious Features の特徴を十分に考慮する必要がある．

Pezeshkpour らは，Spurious Features には *Granular feature* (語彙的素性) と *Abstract feature* (抽象的素性) の 2 種類があると分類している [8]．前者は，“Spielberg” のような予測と無関係な個別の単語である．後者は，単語の重複 (Lexical overlap) のように表層的に現れない高次なパターンのことを指す．

この分類を推論パターンにも適用し，Granular feature をトリガーとする推論パターンを語彙的推論パターン，Abstract feature をトリガーとする推論パターンを抽象的推論パターンとする．なお，抽象的推論パターンについては，この論文では扱わず，今後の課題とする．以後，特に断りがない限り，「推論パターン」あるいは「p」は語彙的推論パターンを指すものとする．

2.1.2 語彙的推論パターンの定義

2.1.1 の分類より，語彙的推論パターンのトリガー t を単語の系列 $\mathbf{w} = \{w_1, w_2, ..., w_n\}$ で表す (式 (2))．

$$p_{granular} \stackrel{\text{def}}{=} \mathbf{w} \xrightarrow{f} \text{Label} \tag{2}$$

単語の系列である理由は，語彙的素性はある特定の一単語ではなく，複数単語の組み合わせ等の多様な形態を持っていると想定し，それに対応するためである．

2.2 推論パターンの候補の抽出

推論パターンを抽出する手法として，新たに Input Reduction (IR) を導入する．IR では，式 (2) におけるトリガー $\mathbf{w}$ について，ラベルの予測に必要となる最低限の単語の系列と考える．また，Label はそれを入力としたときの出力ラベルとする．これは，トリガーがある予測ラベルを引き起こすため**だけ**に用いられた情報であることを保証するためである．

IR の大きな流れは以下の通りである．あるデータセット $\mathcal{D}$ を入力として受け取った後，データセットの各インスタンス (x, y) について，推論パターンの候補 $w = (\mathbf{w}, l)$ を抽出し，その集合 C を出力する．

Algorithm 1 IG による Input reduction の擬似コード

```
function INPUT_REDUCTION_IG(𝒟)
    for all (x, y) ∈ 𝒟 do
        ŷ ← f(x) ; x' ← x
        while ŷ = ŷ' do
            x'_prev ← x' ; x' ← IG_mask(x')
            ŷ' ← f(x')
            if all words in x' are mask then
                break
            end if
        end while
        C ← C ∪ {p = (x'_prev, ŷ')}
    end for
    return C
end function
```

候補の抽出では，入力系列 x が与えられたとき，x に含まれる各単語に対して一つずつマスクをかける ([MASK] に置換する)．マスクの数を増やしていき，それを入力をしたときの予測が変化するまで繰り返す．予測が変われば，その直前の系列をトリガーとする推論パターンの候補を得る．

しかしながら，ナイーブな実装ではマスクの組み合わせが膨大になり計算量が負担となる．したがって，Integrated Gradient (IG)[11] を利用して，予測に対する重要度の低い単語順にマスクをかける (Algorithm 1)．

2.3 一般性の計算

推論パターンを，推論における「パターン」と呼ぶためには，一定の規則性が認められなければならない．そこで，推論パターンの一般性を計算する．

C の i 番目の推論パターン候補 $p_i = (\mathbf{w}_i, l_i)$ の一般性 g_i を次のように計算する．$\mathcal{D}$ とは別のデータセット $\mathcal{D}'$ を用意し，トリガーの単語の系列 $\mathbf{w}_i$ を含む事例の集合 $(x'_j, y'_j) \in E(\mathbf{w}_i)$ を $\mathcal{D}'$ から取得する．このとき，それぞれの事例に対する予測 $f(x'_j)$ がラベル l_i と一致する割合をその推論パターンの候補の一般性とする．形式的には次のように定義する[1)]:

$$g_i \stackrel{\text{def}}{=} \frac{\sum_{x'_j \in E(\mathbf{w}_i)} [f(x'_j) = l_i]}{|E(\mathbf{w}_i)|} \times 100 \tag{3}$$

2.4 Shortcut Reasoning の判定

1 節より，推論パターンが Shortcut Reasoning であることの条件として，(i) IID の入力に対しては有効

1) $[a = b]$ は，$a = b$ のとき 1，$a \neq b$ のとき 0 を返す関数．

に機能するが，(ii) OOD の入力に対しては有効ではないこと の**両方**を満たしていることが挙げられる．

条件 (i) は，IID のデータセット $\mathcal{D}_{\text{IID}}$ より抽出された推論パターンが正しく正解ラベルを予測できていれば，その条件を満たすことになる．条件 (ii) は，OOD のデータセット $\mathcal{D}_{\text{OOD}}$ からトリガー $\mathbf{w}$ に合致する事例の集合 $(x', y') \in E(\mathbf{w})$ を取得し，その予測の多くが誤りであればその条件を満たすといえる．そこで，$E(\mathbf{w})$ と $\mathcal{D}_{\text{OOD}}$ における予測の F1 スコアの差 Δ を計算し (式 (4))，その差を推論パターンがどれだけ OOD での予測に失敗しているかの指標とする．

$$\Delta \stackrel{\text{def}}{=} \text{F1}(E(\mathbf{w}), f) - \text{F1}(\mathcal{D}_{\text{OOD}}, f) \tag{4}$$

したがって，ある推論パターン $p_i = (\mathbf{w}_i, l_i)$ が Shortcut Reasoning であるとは，そのトリガー $\mathbf{w}_i$ に合致する $E(\mathbf{w}_i)$ について Δ_i が小さく，かつラベル l_i に関する条件 (i) を満たすことと定義する．形式的には，Shortcut Reasoning の集合 $\tilde{P}$ は式 (5) のように書ける．

$$\tilde{P} \stackrel{\text{def}}{=} \{p_i = (\mathbf{w}_i, l_i) \in C \mid l_i = y_i, \Delta_i < 0\} \tag{5}$$

ここで，y_i は推論パターン p_i の抽出に用いた IID のデータの正解ラベルである．

2.5 Shortcut Reasoning 検出の手順

手順は大きく分けて 3 つのステップから成る．ステップ 1 では，$\mathcal{D}_{\text{IID}}$ よりモデルの推論パターンの候補 $p_i = (\mathbf{w}_i, l_i) \in C$ を抽出する．ステップ 2 では，抽出された候補の一般性 g_i を $\mathcal{D}_{\text{OOD}}$ を用いて計算し，一般性の高いものを推論パターンとして定義する．最後に，ステップ 3 では，推論パターンのうち，条件 (i, ii) に当てはまるものを Shortcut Reasoning と判定する (式 (5))．

3 実験

3.1 設定

Sentiment Analysis (SA) と Natural Language Inference (NLI) の 2 つのタスクを対象に，提案手法の実験を行う．提案手法が適切に Shortcut Reasoning を検出できているかの検証に既知の情報を必要とするが，これらのタスクは先行研究によって Shortcut Reasoning の形態が報告されていることが採用の背景にある．

今回の実験では，Shortcut Reasoning の基準を $\Delta < -5$ と設定する．また，一般性の低い推論パターンをより正確に排除するために，$|E| \geq 100$ である推論パターンのみを分析の対象とする．

3.1.1 データセットと予測モデル

SA **TweetEval** [12] は Twitter に投稿されたツイートにいくつかの情報がアノテーションされた英語のデータセットである．sentiment はそのサブセットであり，positive/neutral/negative の 3 つの極性クラスがラベル付けされている．**MARC** [13] は Amazon における多言語の商品レビューと 5 段階の星の数による評価がアノテーションされたデータセットである．今回は英語を使う．前処理として，5 段階の評価に関して，星の数が 4 以上のレビューを positive，3 を neutral，2 以下を negative とした．

NLI **MNLI** [14] は，premise と hypothesis の 2 つの文に対し entailment/neutral/contradiction の 3 つのラベルがアノテーションされた NLI のデータセットである．contradiction がラベル付けされている事例の hypothesis の多くに negation が含まれるという Spurious Feature が明らかになっている [1]. **ANLI** [15] は MNLI 同様に 3 種類のラベルがアノテーションされており，MNLI よりも複雑で難易度の高い NLI のデータセットである．

本実験では，IID として TweetEval と MNLI，OOD として，MARC と ANLI を使用する．実験に使用したデータセットの詳細を付録 A に示す．

予測モデルは，Huggingface で公開されている RoBERTa [16] ベースの fine-tuned モデルを使用する．詳細は付録 B に示す．

3.1.2 Input reduction の適用先

Input reduction の入力として，train (学習データ) と test (テストデータ) の 2 通りを用いる．前者はモデルの傾向 (内部状態) を捉えた推論パターンが得られることを期待する．後者は，推論パターンを得るための最も直観的な手法である．本実験では，ランダムに選んだ 1,000 件のデータに IR を適用する．

3.2 結果

検出された Shortcut Reasoning に該当する推論パターンを表 1 に示す．train/test は，推論パターンがそれぞれ train，test に対して IR を適用して抽出されたかを表し，パターンが得られた場合は T，なければ F と記す．

SA 得られた推論パターン全体を見てみると，そ

表 1 検出された Shortcut Reasoning の例

p (SA)	g	Δ	$\|E\|$	train/test	p (NLI)	g	Δ	$\|E\|$	train/test
[“worst”]→ negative	97.5	-24.0	158	T/F	[“/s”, “is”, “popular”]→ neutral	85.3	-9.4	291	T/F
[“Excellent”]→ positive	96.2	-11.9	184	F/T	[“/s”, “never”]→ contradiction	80.9	-7.7	1515	T/T
[“Perfect”]→ positive	96.0	-13.4	324	T/F	[“/s”, “as”, “well”]→ neutral	60.9	-12.0	151	F/T
[“Poor”]→ negative	95.3	-8.8	169	T/T	[“/s”, “not”]→ contradiction	54.5	-22.8	8708	T/T

の多くが [“love”] → positive や [“awful”] → negative 等の感情語に関連したものであった．このことから，SA においてモデルは感情語を予測の重要な手がかりとしていることがわかる．Shortcut Reasoning であるものについても，感情語を含む推論パターンが多かった．したがって，Shortcut Reasoning は必ずしも先行研究で報告されている [“Spielberg”] → poitive のような予測と無関係な単語のみに反応しているのではないことがわかる．一方，そのような予測と無関係な単語がトリガーとなる推論パターンは得られなかったが，IID と OOD を逆にした設定では異なる結果が得られる可能性がある．

MARC に関して，neutral とラベル付けされたレビュー (星 3 つ) においては positive な表現と negative な表現が混ざったものが多く見られた．Δ の絶対値が大きい推論パターンを観察してみると，総じて neutral を誤って予測していることから，本来であればレビューを総合的に評価しなければならないにも関わらず，レビュー中のどちらか一方の極性の感情語だけに依存して推論していることがわかった．

NLI “/s” は入力の premise と hypothesis を隔てる目印である．得られた推論パターンを概観してみると，hypothesis に含まれる単語がトリガーの大半を占め，premise の中の単語を含む推論パターンは数件しかなかった．この結果はモデルが hypothesis に依存して文間関係を予測している[2)]ということを意味し，同様の結果が Poliak 等によって報告されている [2]．さらに，否定表現が hypothesis に含まれている推論パターンが多く Shortcut Reasoning として判定されたが，これについても先行研究 [1] で指摘されている Shortcut Reasoning である．以上から，提案手法が多様な Shortcut Reasoning を適切に検出できていることがわかった．

train or test Input reduction の適用先について，train と test から得られた推論パターンには大きな差は見られなかった．具体的なパターンは異なるが，特徴はおおむね同じであった．

未知の Shortcut Reasoning と考えられるもの NLI では hypothesis に含まれる [“popular”] → neutral や，[“as”, “well”] → neutral が新しい Shortcut Reasoning として得られた．どちらも一般性，スコア差ともに十分に Shortcut Reasoning と判断できる水準にある．一方，SA については特に見つけることができなかった．

4 関連研究

Wang らは，事前の定義なしに自動的に Shortcut Reasoning を検知する手法を提案している [10]．この研究では出力ラベルの予測に有効な特徴（[“good”] → positive, [“bad”] → negative 等）は Shortcut Reasoning になりえないとし，それらを排除した上で明らかに予測に関係のない特徴（[“Spielberg”] → positive 等）のみを検知しようとしている．しかしながら，Joshi らによると，自然言語処理のタスクにおいて Shortcut Reasoning の原因となる特徴（Spurious Features）は，大半が予測のための有力な情報であることを示しており [17]，このような特徴を排除することは適切とは言えない．さらに，2.3 項で述べた推論パターンの一般性を考慮していないため，検出された Shortcut Reasoning が実際にどの程度予測に影響を与えているのかわからないという問題点を持っている．

5 おわりに

本研究では，モデルの推論パターンを新たに定義し，Shortcut Reasoning を検知する手法を提案した．実験結果では，Shortcut Reasoning の形態についての最小限の定義で，先行研究で明らかになっていた Shortcut Reasoning に加え，少量ながら未知な新しいものを自動的に検知することに成功した．今回設定した推論パターンは語彙的推論パターンにのみ対応しており，抽象的推論パターンへの対応は今後の課題である．さらに，抽出型の機械読解等のより複雑なタスクへの対応や，この手法で得られた Shortcut Reasoning の情報を応用し，モデルの頑健性を向上させることにも取り組みたい．

2） なお，この Shortcut Reasoning は 2.1.1 で定義した抽象的推論パターンに分類されるものである．

謝辞

本研究は JSPS 科研費 19K20332 の助成を受けたものです.

参考文献

[1] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation Artifacts in Natural Language Inference Data. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[2] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis Only Baselines in Natural Language Inference. In **Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics**, pp. 180–191, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[3] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics.

[4] Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. Beyond Leaderboards: A survey of methods for revealing weaknesses in Natural Language Inference data and models, May 2020. arXiv:2005.14709 [cs].

[5] Xuezhi Wang, Haohan Wang, and Diyi Yang. Measure and Improve Robustness in NLP Models: A Survey. In **Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 4569–4586, Seattle, United States, July 2022. Association for Computational Linguistics.

[6] Xanh Ho, Johannes Mario Meissner, Saku Sugawara, and Akiko Aizawa. A Survey on Measuring and Mitigating Reasoning Shortcuts in Machine Reading Comprehension, September 2022. arXiv:2209.01824 [cs].

[7] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics.

[8] Pouya Pezeshkpour, Sarthak Jain, Sameer Singh, and Byron Wallace. Combining Feature and Instance Attribution to Detect Artifacts. In **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 1934–1946, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[9] Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. Explaining Black Box Predictions and Unveiling Data Artifacts through Influence Functions. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 5553–5563, Online, July 2020. Association for Computational Linguistics.

[10] Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. Identifying and Mitigating Spurious Correlations for Improving Robustness in NLP Models, May 2022. arXiv:2110.07736 [cs].

[11] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks, June 2017. arXiv:1703.01365 [cs].

[12] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 1644–1650, Online, November 2020. Association for Computational Linguistics.

[13] Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. The multilingual Amazon reviews corpus. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 4563–4568, Online, November 2020. Association for Computational Linguistics.

[14] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[15] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics.

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[17] Nitish Joshi, Xiang Pan, and He He. Are All Spurious Features in Natural Language Alike? An Analysis through a Causal Lens, October 2022. arXiv:2210.14011 [cs].

[18] Twitter-roBERTa-base for sentiment analysis, 2023-1 閲覧. https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment.

[19] roberta-large-mnli, 2023-1 閲覧. https://huggingface.co/roberta-large-mnli.

A　データセットの詳細

データセット	train	validation	test
SA			
Tweeteval (sentiment)	45,615	2,000	12,284
MARC (en)	200,000	5,000	5,000
NLI			
MNLI (matched)	392,702	9,815	9,796
ANLI (round3)	100,459	1,200	1,200

B　使用したモデルの詳細

タスク	モデル
SA[18]	cardiffnlp/twitter-roberta-base-sentiment
NLI[19]	roberta-large-mnli