

イベントの発生条件のアノテーションと条件の予測性能評価

市村 裕章 三輪 誠 佐々木 裕
豊田工業大学

{sd20015,makoto-miwa,yutaka.sasaki}@toyota-ti.ac.jp

概要

本論文では、イベントとそのテキスト内の条件の言及を紐付ける手法を提案する。テキストからの情報抽出では、テキスト中のイベントをあらかじめ決められたタイプに汎化して、構造化して抽出する。しかし、イベントはそのテキスト中の特定の条件において起こせるものであり、そのイベントがいつでも起こせるわけではない。本研究では、イベントが起こせる条件を明確化することを目的として、テキスト内のイベントの発生条件の言及をイベントと紐付けるアノテーションを提案する。さらに、アノテーションしたデータを用いて、テキスト内の対象のイベントに対してテキスト内の条件言及を自動検出する手法を提案・評価する。実験では日本の交通ルールのデータセットにおいて、高いアノテーション一致率を達成したが、条件の予測性能は低い結果となった。

1 はじめに

テキストからの情報抽出では、テキスト中の用語とイベントの関係を汎化して構造化して抽出する。例えば、日本の交通ルールのデータセットである OSR-RoR (Ontology Style Relation - Rule of the Road) [1] では、「運転者はシートベルトを着用すること」に対して、「運転者」を「Driver」、「シートベルト」を「SeatBelt」、「着用」を「Utilization」のタイプに分類し、「Driver」を主語・「SeatBelt」を目的語とした「Utilization」イベントとして、汎化・構造化して抽出する。

しかし、この汎化・構造化して抽出したイベントはテキスト中の特定の条件において起こせるものであり、そのイベントをいつでも起こせるとは限らないという問題がある。先述した「Driver」・「SeatBelt」・「Utilization」は、「自動車を運転するとき」という条件の中で出現した用語を汎化したものである。そのため、「二輪車を運転するとき」とい

う条件の中で出現した「運転者」を同じ「Driver」の用語タイプで汎化したとしても、「自動車を運転するとき」の「SeatBelt」を対象に「Utilization」のイベントを起こすことはできない。このように、抽出したイベントを利用するにはイベントの発生条件を考慮しなければならない場合がある。

上記の問題の対応としては、イベントの発生条件を考慮して「Driver」をはじめとする用語タイプ自体をより細分化してタグ付けし直す方法が考えられる。しかし、この方法では、条件ごとに用語タイプを用意する必要がある、あらかじめ用語タイプを定義することができない。また、複数の条件が重なった際に初めて起こすことができるイベントなども考えると、用語タイプが複雑になってしまい、タグ付け作業のコストが高くなってしまふ。

そこで、本研究では、イベントの条件を明確化することを目的として、イベントとテキスト内の条件の言及を紐付けるアノテーションを提案する。本稿では、アノテーション手法の検証のために、OSR-RoR の「Driver」の用語タイプを主語としたイベントを対象に、イベントの発生条件をタグ付けしたデータセットを作成する。また、イベントの発生条件の抽出を自動化するため、ある特定の表現がテキスト内のどの範囲に影響を及ぼすか検出するタスクであるスコープ検出を応用して、イベントに対して、そのテキスト内の条件の言及を検出する方法も提案する。

本研究の貢献は以下の通りである。

- イベントとそのテキスト内の条件の言及を紐づけることで、イベントの発生条件を明確にするアノテーションを提案し、高い一致率を達成可能なアノテーション基準を設計した。
- OSR-RoR の「Driver」の用語タイプを主語としたイベントを対象にイベントの発生条件をタグ付けしたデータセットを作成した。
- イベントの発生条件の抽出を自動化するため、

イベントに対して、そのテキスト内の条件の言及を検出する手法を提案したが、予測性能は低い結果となった。

2 関連研究

2.1 OSR-RoR

Savong らにより作成された OSR-RoR は、全 11 章からなる日本の交通ルールを記載した日本語テキストに対して、アノテーションを行なったデータセットである。OSR-RoR は、知識をコンピュータ上で表現したオントロジーの構築に利用するため、用語タイプ間の関係ラベルがオントロジーの形式で付けられている。具体的には、オントロジーの形式に合わせて、主語、述語、目的語の関係がトリプルとして汎化して抽出できる形でタグづけされている。しかし、抽出対象のトリプルにはその発生条件が考慮されていないため、その主語に対応する用語タイプの用語が述語を起こせるとは限らない、述語が目的語に対応する用語タイプの用語を取れるとは限らない、などの問題がある。

2.2 文・節間の談話関係のアノテーション

文・節間で条件を関係付けを行っているアノテーションとして、談話関係のアノテーション [2, 3] がある。このアノテーションは、イベントの発生条件が隣接した文・節に記述されたイベントの発生条件は表現可能であるが、談話構造の解析を目的としているため、離れた文・節やイベントごとの発生条件は対象としていない。

2.3 スコープ検出

スコープ検出とは、否定や推測などを示す語をキューとして与え、その語によってテキスト内で影響を受ける範囲（スコープ）を検出するタスクである。スコープ検出は、ルールベースや機械学習による手法などで取り組まれてきたが、近年では BERT (Bidirectional Encoder Representations from Transformers) [4] をベースにしたモデルが高い性能を達成している [5, 6]。本研究では、イベントの発生条件の検出を自動化するため、テキスト内の条件の言及の検出にスコープ検出の手法を応用する。

3 発生条件のアノテーション

3.1 アノテーション方法

イベントを起こせる条件を明確にするために、段落内で条件を言及している箇所をタグ付けし、イベントの発生条件としてイベントに関係付ける。発生条件のアノテーションの流れを図 1 に示す。条件の言及を探す範囲は、テキスト全体とすると際限がないため、段落内に限定する。また、アノテーションでタグ付けする条件の言及の種類を表 1 に示す。1 つのイベントに複数の発生条件がつく場合、そのイベントは、すべての発生条件を満たした上で起こせるイベントとなる。イベントの発生条件をタグ付けする手順は、以下のようになる。

1. 対象のイベントを起こせる条件となる他のイベント、もしくは、時間、場所、天候などを意味する名詞句を同じ段落内のテキストから見つける
2. 条件を言及している他のイベントを含む句や名詞句に条件タグを付ける。ただし、イベントを含む句には直接の項や修飾語を含める。
3. 対象のイベントから条件タグに向かって紐付ける

同じ段落内に複数の文がある場合、前の文を踏まえて後の文で条件の言及が省略されることがある。その場合は、内容を考えて、前の文から条件の言及を紐づけることとする。ただし、条件の言及を章タイトルから採用することはしない。また、同じ条件の言及がテキスト内に複数存在する場合は、直前のものを使用する。

並列で記述された複数の条件のうち片方の条件を満たせば良い場合は、複数の条件の言及を一括りにしてタグ付けする。例えば、「ドアを開けるときや車から降りるときは、他の車に注意しましょう」という記述では、注意のイベントの発生条件は「ドアを開けるとき」もしくは「車から降りるとき」である。このとき、両方の条件の言及にそれぞれタグ付けし、イベントに関係付けてしまうと、両方の発生条件を満たさないと起こせないイベントとなってしまう。よって、このような場合は、「ドアを開けるときや車から降りるとき」までを一括りにして条件タグを付ける。

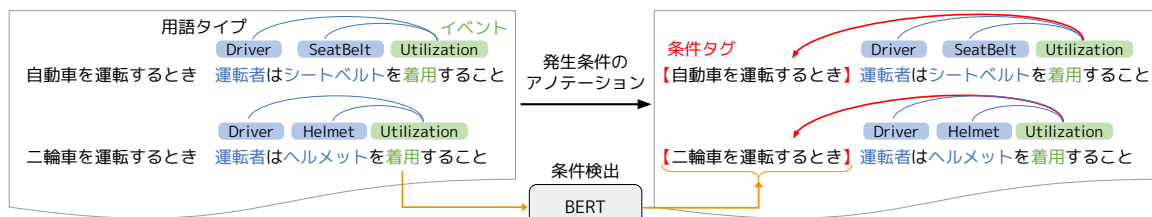


図1 提案手法の流れ

表1 条件の言及の種類

条件の言及	例 (太字がイベント, 下線部が条件の言及)
対象のイベントの前に 起こすべき他のイベントを含む句	追い越しをするときは, 運転者は追越車線の車の動きなどに 注意 しましょう
時間, 場所, 天候などを意味する名詞句	雨の日では, 運転者は速度を 落 としましょう

表2 「Driver」を主語に持つイベントのうち, 発生条件を関係付けたイベントの数

章	イベントの数	発生条件を関係付けた イベントの数
1	12	4
2	2	2
3	3	0
4	109	26
5	506	307
6	142	103
7	102	71
8	89	33
9	55	23
10	96	56
11	63	26
合計	1,179	651

表3 イベント毎に関係付けた発生条件の数
発生条件の数 イベントの数

発生条件の数	イベントの数
0	528
1	332
2	199
3以上	120
合計	1,179

また, 発生条件のアノテーションでは実現できないこととして, 章タイトルなど段落外にしかない条件やテキスト内にない暗黙的な条件があることが分かった。例えば, 7章2節で追い越しの仕方を説明した「進路を戻すときは, 運転者は追い越した車全体がルームミラーに映ったことを**確認**してから行いましょう」という文では, 車を運転していることを前提にしているが, 段落内に車を運転している記述が存在しなかった。発生条件のアノテーションは, 段落内に出現する条件の言及と紐づけるため, 条件が段落外にしかない場合や暗黙的な前提である場合に対応できないことが分かった。

次に, イベント毎に関係付けた発生条件の数を表3に示す。この結果より, 複数の発生条件が付いたイベントが多くあることが分かる。このようなイベントは, 用語タイプ自体を細分化してタグ付けし直すと複雑になってしまうため, 条件の言及と紐づけることが有効である。

3.2 アノテーションの結果と考察

OSR-RoR 中で「Driver」を主語にしたイベントを対象に発生条件のアノテーションを行った。タグ付け作業には, ガイドラインを熟知した2人のアノテータが携わり, アノテーションツールである brat rapid annotation tool [7] を使用した。「Driver」を主語にした全イベントのうち, 発生条件をタグ付けたイベントの数を表2に示す。

発生条件のアノテーションにより, 「Driver」を主語にしたイベントのうち, 約半数のイベントに発生条件を関係付けできた。発生条件を関係付けなかったイベントは, そもそも発生条件のないイベントであった。

3.3 アノテーションの評価

発生条件のアノテーションガイドラインが妥当であることを示すため, 2人のアノテータによるアノテーション一致率を測定した。OSR-RoR データ

表 4 検出に失敗した条件の言及を含む文

	条件タグが付いた文 (太字がイベント, 下線部が条件の言及)
正解	<u>交通事故が起きたときは</u> , 運転者や乗務員は事故の続発を防ぐため, 他の交通の妨げにならないような安全な場所に車を止め, エンジンを切らなければなりません.
予測	<u>交通事故が起きたときは</u> , 運転者や乗務員は事故の続発を防ぐため, 他の交通の妨げにならないような安全な場所に車を止め, エンジンを切らなければなりません.

セットのうち, 2人のアノテータがアノテーションしたことがない「Driver」を主語にしたイベントを53件を抜き出し, 発生条件のアノテーションをする. アノテーション一致数は, 片方のタグ付け結果を正しいアノテーションとして, イベント毎にタグ付けした用語とその範囲が一致したものをカウントする. 集計の結果, アノテータ間の一致数は53件中43件で, 一致率は81.1%を達成した.

アノテーションの不一致の原因は, 対象のイベントの前に他のイベントがある場合に, 前のイベントが発生条件となるか曖昧なケースが存在することである. 例えば, 「シートベルトのバックルの金具を確実に差し込み, 固定しましょう」という文で, 固定のイベントの発生条件をタグ付けする場合, 「確実に差し込む」というイベントを固定のイベントの発生条件とするかどうかである. 「確実に差し込む」が「固定される」と同じ意味であると解釈する場合は, 発生条件にしなくても良いと考えられる.

4 条件検出

4.1 条件の言及の検出方法

イベントの発生条件のアノテーションを自動化するため, スコープ検出の手法 [6] を応用して, テキストと対象のイベントから発生条件となる条件の言及を検出する. まず, 対象のイベントを CUE タグで囲った1文のテキストをトークンで分割し, BERT に入力する. BERT の出力では, 条件の言及かを判別するため, トークン毎に BIO (Beginning, Inside, Outside) タグで分類する. 「Driver」の用語タイプを主語に持つイベント毎に発生条件となる条件の言及を検出することにより, イベントの発生条件の検出を自動化できる.

4.2 予測性能評価

発生条件のアノテーションをした OSR-RoR で BERT を学習し, 元の OSR-RoR のテキストから対象のイベントを含んだ1文のみを入力して, 条件の言

及を予測できるか評価した. 発生条件のアノテーションをした OSR-RoR のうち, 1章から5章までを訓練データ, 6章から7章までを開発データ, 8章から11章までをテストデータとした. BERT には, 日本語に対応した cl-tohoku/bert-base-japanese-v3 [8] を使用した. 評価は F-score で行い, 入力文よりも前の文に出現する検出できない条件の言及を全て負例として扱った.

条件検出の結果, 開発データでは 39.3%, テストデータでは 23.0% となり, 両方の結果とも低い値となった. この理由としては, 訓練データの件数が 500 件程度と少ないこと, 文を跨いだ条件の言及を検出できないことが考えられる.

検出した条件の言及を実際に確認したところ, 表 4 に示すように, 長い条件の言及に対して検出に失敗した例が散見された. この文では, 「交通事故が起きたとき」の検出に成功した, 一方, 「他の交通の妨げにならないような安全な場所に車を止め」の検出に失敗した. そのかわり, 「他の交通の妨げ」と「安全な場所に車を止め」に分割して検出してしまっていた.

5 おわりに

本研究では, イベントの条件を明確化することを目的として, イベントとテキスト内の条件の言及を紐付けるアノテーションを提案した. OSR-RoR 内で「Driver」の用語タイプを主語にしたイベントを対象に, 高いアノテーション一致率を達成可能なアノテーション基準を設計し, 発生条件のアノテーションをしたデータセットを作成した. さらに, イベントの発生条件の抽出の自動化に向けて, テキスト内の条件の言及を検出する実験を行ったが, 予測性能は低い結果となった.

今後は, 全イベントを対象にした発生条件のアノテーションを検討する. また, BERT に入力する文の数を増やすなど, 条件検出の性能を向上させる方法についても模索する.

謝辞

本研究は JSPS 科研費 JP23K11237 の助成を受けたものです。

参考文献

- [1] Savong Bou, Makoto Miwa, and Yutaka Sasaki. Two evaluations on ontology-style relation annotations. Computer Speech & Language, Vol. 84, p. 101569, 2024.
- [2] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltasakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors, Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).
- [3] W. C. Mann and S. A. Thompson. Rhetorical structure theory. Toward a functional theory of text organization., Vol. Text, 8, 3. 1988.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [5] Aditya Khandelwal and Suraj Sawant. NegBERT: A transfer learning approach for negation detection and scope resolution. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 5739–5748, Marseille, France, May 2020. European Language Resources Association.
- [6] Priya Tiwary, Akshayraj Madhubalan, and Amit Gautam. No means 'No': a non-improper modeling approach, with embedded speculative context. Bioinformatics, Vol. 38, No. 20, pp. 4790–4796, August 2022.
- [7] brat rapid annotation tool. <https://brat.nlplab.org/> (12月29日アクセス) .
- [8] cl-tohoku/bert-japanese: Bert models for japanese text. <https://github.com/cl-tohoku/bert-japanese/> (12月26日アクセス) .
- [9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, Vol. 32. Curran Associates, Inc., 2019.
- [10] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac,

Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45, Online, October 2020. Association for Computational Linguistics.

A ハイパーパラメータ

本研究における実験で用いたハイパーパラメータを表 5 に示す.

表 5 モデルのハイパーパラメータ

ハイパーパラメータ	値
学習率	2×10^{-5}
重み減衰	1×10^{-4}
エポック数	4
バッチサイズ	2

B 実験環境

実験のためのプログラミング言語は Python 3.7.11, 機械学習ライブラリとして PyTorch [9] のバージョン 1.10.0, 事前学習モデルを使用するために Transformers [10] のバージョン 4.25.1 を用いた. また, 計算機では CPU に Intel(R) Core(TM) i7-5960X CPU @ 3.00GHz, GPU に GeForce GTX TITAN X を用いた.